JISC Authentication, Authorisation and Accounting (AAA) Programme

Technologies for Information Environment Security: TIES project report

CONTENTS

EXECUTIVE SUMMARY IV

SECTION I – INTRODUCTION 1
1 Scope 1
2 Partners 2
3 References 2
4 Abbreviations 3

SECTION II – DESIGN ISSUES 4
5 Model 4
6 General requirements 4
6.1 Applications of the ac-PKI 4
6.2 Simplicity 5
6.3 Standards-based 5
6.4 Application in FE 6
6.5 User confidentiality 6
6.6 Service environments 6
7 Stakeholders 7
7.1 Users 7
7.2 Institutions 7
7.3 DSPs 8
8 Risk analysis 9
8.1 Assets 9
8.2 Threats 9
8.3 Vulnerabilities 10
8.4 Institutional applications 11
8.5 Grid applications 11
8.6 Risk assessment 11
8.7 Key theft and disclosure 11
9 Estimates 12
10 Authorisation 13
11 Technical options 13
11.1 Key generation 13
11.2 Revocation 15
11.3 Certificate and CRL profiles 17
11.4 Use of Directory 19
12 PKI architecture 19
13 Existing institutional PKIs 20
14 Policy requirements 21
14.1 Policy classes 21
14.2 Grid 22
14.3 US HE model certificate policies 22
14.4 Two-tier policies 23
14.5 Policies for ac-PKI 23
15 Total cost of ownership (TCO) 24

SECTION III – CONCLUSIONS AND RECOMMENDATIONS 25
16 Conclusions 25
16.1 Initial questions 25
16.2 Effectiveness of X.509 27
16.3 CA architecture 27
16.4 Procurement 28
16.5 Policy 29

16.6 Technical issues 30
16.7 Preconceptions 31
17 Recommendations 31
17.1 Adoption of technology 31
17.2 CA architecture 32
17.3 Vendor evaluation 32
17.4 Deployment plan 32
17.5 Engagement 33

ANNEX A – TECHNICAL REALISATION 34
A.1 Overview 34
A.2 Institutional RA 34
A.3 Unix CA 36
A.3.1 Component identification 36
A.3.2 System design 38
A.3.3 System integration 39
A.3.4 Import of user information from the RA 40
A.3.5 Requesting a certificate 40
A.3.6 Administration 41
A.3.7 Audit 42
A.4 Microsoft CA 42
A.4.1 Installation 42
A.4.2 Web-based certificate enrolment 43
A.4.3 Granting certificates 43
A.4.4 Certificate compatibility 44
A.4.5 Certificate revocation 46
A.4.6 Microsoft CA conclusions 46
A.5 Unix DSP certificate verifier 47
A.5.1 Chosen implementation 47
A.5.2 Configuration 47
A.5.3 Integration with data services 48
A.5.4 DSP audit 51
A.6 Microsoft DSP certificate verifier 51
A.6.1 Installing the web server certificate 52
A.6.2 Requiring client authentication 52
A.6.3 Trusted root certification authorities 52
A.6.4 Certificate trust lists 53
A.6.5 Handling of revoked certificates by IIS 53
A.6.6 Audit 54
A.6.7 Microsoft DSP conclusion 54
A.7 DSP authorisation package 54
A.7.1 Shibboleth 54
A.7.2 Athens Migration 56
A.7.3 Licensing Authority Tables 57
A.8 Browser support 58
A.9 Revocation 60
A.9.1 CRL handling at the DSP 60
A.9.2 OCSP handling at the CA 60
A.9.3 OCSP handling at the DSP 61
A.9.4 Possible revocation schemes 61

ANNEX B –SIMPLE REGISTRATION MODEL 63
B.1 Registration Authority activity 63
B.2 User/CA interaction 65
B.3 Recovery 65

EXECUTIVE SUMMARY

The TIES Project was funded by the JISC to consider the problem of *authentication* in the UK academic environment. Within the JISC Authentication, Authorisation and Accounting (AAA) Programme, the task of TIES has been to investigate the technical, managerial, and practical issues in the deployment of X.509 digital certificates within a Public Key Infrastructure to be developed for the UK academic community (ac-PKI).

There are three distinct areas where the ac-PKI could address authentication requirements in the community:

a) *The JISC Information Environment*. It could replace the present proprietary Athens access management scheme with an industry-standard technology for authentication.

b) *Reduced signon within institutions*. Many institutions are seeking solutions to reduce the number of separate logins required by users for local services. PKI provides reduced (effectively zero) signon for institutional web-based services.

c) *The UK e-Science Grid*. The present method of certification for the Grid will not scale to accommodate the expected levels of growth, and a new PKI framework may be required.

The crucial advantage of a PKI-based solution is that this technology is already deployed in all mainstream browsers found on user desktops, and in the main web servers in use in the community. The infrastructure to support the ac-PKI must still be provided, but will have immediate application and will not require substantial development. Moreover, no other technology offers a unified solution for all three service areas listed above.

To understand the practical problems of deployment of the ac-PKI, TIES implemented and tested each of its key components: Certification Authority, Registration Authority, certificate repository, certificate verifier, and mainstream browsers. In all cases, both open source and Microsoft implementations were found to be serviceable. In the case of the institutional Certification Authority, however, the solution though successful as a technical exercise did not satisfy the organisational, management, and human factor requirements relevant for a national authentication service. A vendor-supplied Certification Authority is required.

Further, institutions naturally fulfil the role of Registration Authority, which mirrors the existing administrative processes for user induction and assignment of local credentials. By removing the Certification Authority function from the institution to a dedicated authority, the cost of participation in the ac-PKI is considerably reduced, and the possibility of deployment across the whole sector becomes a more realistic proposition.

With a subject of this complexity, different conclusions may be drawn from the findings presented in the body of the report. Having weighed the evidence, this report has reached the conclusions presented in section III, and makes the following recommendations:

a) Adopt the ac-PKI as the preferred unified solution for authentication in the JISC IE, in the e-Science Grid, and for local institutional services.

b) Base the model for the ac-PKI on a vendor-supplied central Certification Authority, with a set of institutional Registration Authorities.

c) Undertake further evaluation of commercial PKI products to determine the relative costs of insourced and outsourced solutions, and the quality of support for institutional RAs.

d) Plan a staged deployment of the ac-PKI, with a first stage to identify any issues specific to UK use and provide roadmaps for other institutions to follow.

e) Engage with institutions, and with service providers in the ac-PKI, including JISC-funded centres, commercial publishers, EduServe, e-Science Grid, and the Common Information Environment.

Given the pace of technological change, the case can always be made for postponement of major decisions such as for the deployment of a national service. In the present case of an authentication service, however, there are few reasons to adopt the Micawber expectation of something (better) turning up. The penalty for delay will be the continuing fragmentation of authorisation policy in the UK, and the proliferation of bespoke solutions.

© EDINA, University of Edinburgh 2003

SECTION I – INTRODUCTION

The JISC Authentication, Authorisation and Accounting (AAA) Programme was established to investigate the various technical, management, and pragmatic issues for the use of next-generation technologies for authentication and authorisation in the UK H&FE sector. The Programme will address the emerging requirements arising from developments in the broader education and research community.

The following specific requirements have been identified:

- for a next generation solution for authentication and authorisation, relevant over the five year horizon;
- based on open, vendor-independent standards, effective for international use;
- a strategic replacement for the present Athens service;
- relevant to the wider requirements of the education community, including local institutional use, national and international collaboration (e-Science Grid), as well as the JISC Information Environment (IE);
- capable of co-existence with current access management technologies.

The TIES project, which is led by EDINA, has investigated *authentication* based on the use of X.509 digital certificates within a Public Key Infrastructure model for the UK academic community (ac-PKI).

X.509 identity certificates [5] represent a significant industry-standard technology for authentication in commercial and governmental sectors, though its use in the academic community has mainly been limited to server-level authentication and the e-Science Grid.

This development of a PKI is a complex undertaking, and the aim of TIES, therefore, has been to investigate the practicality of this deployment by implementing the full range of PKI components in a UK academic setting, and to consider the PKI design options most effective for use in UK H&FE.

Section I contains an introduction. Section II considers design issues, including the model, service requirements, risks, policy issues, and technical options. Section III contains conclusions and recommendations. Annexes A and B describe the technical realisation of the various organisational and software components required to construct a PKI.

1 Scope

The project has demonstrated practical deployment of digital certificates within an HE institution and their use for authenticating users to a data service provider (DSP) in the JISC IE. This has involved the implementation of a functional Public Key Infrastructure as a demonstration of proof of concept. The following tasks were undertaken:

- implementation of an institutional Registration Authority;
- implementation of a Certification Authority, using Unix/Apache server technology;
- implementation of a Certification Authority, using Microsoft server technology;
- implementation of certificate verification in a standard JISC IE service based on Unix/Apache;
- implementation of certificate verification in an institutional Microsoft-based service;
- testing certificate handling in the mainstream browsers for standard desktop equipment (Windows, Unix/Linux, Mac OS);
- evaluation of the major design issues for an academic PKI (risk assessment, revocation, key generation, architecture, policy, costs);
- recommendations for a national ac-PKI solution.

The scope of application of the ac-PKI is limited to user authentication for web-based services accessed using standard browsers. This excludes secure (encrypted) email, access to password-based services (telnet, pop, imap), and advanced security services such as non-repudiation.

The focus of the project has been authentication, and has not included a comprehensive evaluation of authorisation technologies. Some specific authorisation technologies have been implemented, however, to prove the integration of PKI authentication with a range of authorisation methods.

While the initial motivation for an authentication service is to provide access management for the licensed resources of the JISC IE, the other major application areas of local institutional services and e-Science Grid are also considered.

2 Partners

In a project of this type, concerning major infrastructure issues affecting the whole academic sector, it is valuable to solicit the views of other organisations representative of the mixed economy of HE, FE and commercial data service providers (DSPs). The project brought together associate partners of various types:

- the University of Edinburgh (the lead institution, representing an ‘old’ University);
- Napier University (a ‘new’ University);
- Stevenson College (a large FE institution);
- Newark and Sherwood College (a medium-sized FE institution);
- the Institute of Physics Publishing (a commercial data service provider).

Different groups within the University of Edinburgh made contributions to the project:

- EDINA, a JISC-funded national DSP, provided DSP expertise and project and technical management;
- Computing Services provided technical expertise on integration with University systems;
- Registry and Human Resources provided the core user record data.

Professor D.W.Chadwick, University of Salford provided expert advice on the application of X.509 technology.

3 References

[1] *Bridge Certification Authorities: Connecting B2B Public Key Infrastructures,* William T. Polk and Nelson E. Hastings, NIST, http://csrc.nist.gov/pki/documents/B2B-article.doc

[2] *HEPKI Model Campus Certificate Policy*, October 2002, http://middleware.internet2.edu/certpolicies/

[3] *Global Grid Forum Certificate Policy Model*, Randy Butler, Tony J. Genovese, October 2002, http://caops.es.net/Documents/LastCALL/GGF-CPFinal.pdf

[4] ISO 3166-1:1997, *Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes*

[5] ITU-T Recommendation X.509 (2000) | ISO/IEC 9594-8 : 2001, *Information technology – Open Systems Interconnection – Public-Key and Attribute Certificate Frameworks.*

[6] *Public Key Infrastructure Study: Final Report,* Shimshon Berkovits, Santosh Chokhani, Judith A. Furlong, Jisoo A. Geiter, and Jonathan C. Guild. http://csrc.nist.gov/pki/documents/mitre.ps, Produced by the MITRE Corporation for NIST, April 1994.

[7] *Registration and Authentication, e-Government Strategy Framework Policy and Guidelines,* http://www.e-envoy.gov.uk/assetRoot/04/00/09/60/04000960.pdf

[8] RFC 2251, *Lightweight Directory Access Protocol (v3),* December 1997.

[9] RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*.

[10] RFC 3280, *Internet X.509 Public Key Infrastructure, Certificate and Certificate Revocation List (CRL) Profile*

[11] *Scoping study for interim DNER authentication developments*, JISC Committee for Authentication and Security (JCAS), Sandy Shaw, EDINA, September 2000.

[12] TCO for PKI white paper, Verisign, February 2002, http://www.verisign.com/resources/wp/pki/tco/tco.pdf

[13] *The SSL Protocol Version 3.0*, Internet Draft, 1996, http://wp.netscape.com/eng/ssl3

[14] *X.509 Certificate Policy for the Federal Bridge Certification Authority (FBCA)*, http://csrc.nist.gov/pki/fbca/FBCA_CP_20001227.doc

4 Abbreviations

The following abbreviations are used:

AAA	JISC Authentication, Authorisation and Accounting Programme
ac-PKI	Academic PKI (for UK H&FE institutions)
BCA	Bridge Certification Authority
CA	Certification Authority
CIE	Common Information Environment
CMC	Certificate Management Messages over CMS (RFC 2797)
CMP	Certificate Management Protocols (RFC 2510, updated by draft-ietf-pkix-rfc2510bis)
CMS	Cryptographic Message Syntax (RFC 2630)
CREN	Corporation for Research and Educational Networking
CRL	Certificate Revocation List
CP/CPS	Certificate Policy/Certification Practices Statement
DSP	Data Service Provider
FBCA	Federal Bridge Certification Authority
GGF	Grid Global Forum
HR	Human Resources
IIS	Microsoft Internet Information Server
IE	Information Environment
JISC	Joint Information Systems Committee
LDAP	Lightweight Directory Access Protocol (RFC 2251)
NIH	National Institutes of Health
OCSP	Online Certificate Status Protocol (RFC 2560)
PKI	Public Key Infrastructure (X.509)
RA	Registration Authority
SSL	Secure Sockets Layer
TCO	Total cost of ownership
VLE	Virtual Learning Environment
VO	Virtual Organisation

SECTION II – DESIGN ISSUES

5 Model

A PKI may take a variety of forms, but essentially involves a Certification Authority (CA), a Registration Authority (RA), a user, a repository, and a relying party:

a) The user applies to the RA for registration, and provides proof of identity according to a specified level of assurance.

b) The RA transmits registration details of the user to the CA.

c) Once registered, the user applies for certification (either directly to the CA or indirectly via the RA); if granted, the certificate is issued to the user.

d) The certificate is deposited in a repository.

e) The user presents the certificate to a relying party which verifies its authenticity before determining whether the user is authorised to access the requested resource.

TIES has considered the range of factors affecting the design of a PKI for H&FE, but four initial questions stand out:

a) *Can an institution fulfil the RA function by means of its existing registration procedures?*

   A potentially costly aspect of PKI operation is user registration. This would be considerably reduced if an institution, acting as an RA, was able to simply reuse existing administrative database resources already in place for staff and student record management,

b) *Can an institution implement its own CA at reasonable cost using open source or commercial components?*

   This is the approach being followed in a number of institutions in the US (and by the UK e-Science CA, for example). Before promoting this as a preferred solution in the UK, the advantages and disadvantages should be better understood.

c) *Can a service in the JISC IE implement certificate validation using open source or commercial components?*

   Certificates become useful only where they provide relying parties with verifiable evidence of user identity. The question, therefore, is whether the tools exist that would enable JISC IE service providers to implement certificate validation as an alternative to Athens access management.

d) *Is certificate handling in the mainstream browsers for each of the common desktop platforms both functional and useable?*

   Browsers have offered some support for digital certificates for several years, but it is to be confirmed that the level of support is adequate for ac-PKI requirements.

6 General requirements

The ac-PKI must satisfy the various general requirements described below.

6.1 Applications of the ac-PKI

There are three areas where the ac-PKI has potential application:

a) *The JISC Information Environment*. This provides the immediate motivation for the development of an ac-PKI, arising from the desire to migrate from the Athens Access Management scheme (http://www.athens.ac.uk/) to an open, standards-based solution.

b) *Institutional applications*. Increasingly, institutions are developing web-based services for their users for the presentation of a wide range of applications, such as course and exam details, Student Association and College announcements, Library services, and VLEs. PKI technology is already deployed in all mainstream web browsers and could potentially provide authentication for these web-based services.

c) *The Grid*. From the outset, authentication for Grid services has been based on PKI technology. The UK National e-Science CA supports a dedicated PKI for a modest number of users. The CA operates a 'medium assurance' certification regime which is relatively costly to administer and is unlikely to scale much beyond the present level of deployment. Future growth in the use of Grid services will require corresponding growth in the certification infrastructure.

6.2 Simplicity

The successful deployment of a PKI for the academic community depends upon attracting support from the various parties involved, principally institutions, DSPs, and users.

For institutions, the problem is the organisational task of managing the processes used to assign digital certificates to their users. This is complicated by a number of factors:

— there are more than 400 institutions which vary substantially in character;

— while some institutions enjoy the advantage of a fully-staffed technical capacity, others operate with a more limited technical resource;

— institutions may regard PKI as a heavyweight security solution for simple access to JISC IE services;

— the use of PKI as a practical method for authentication to local web-based services is not widely realised.

For DSPs, the problem is the technical implementation of certificate verification. This problem makes some technical demands, but is reasonably bounded:

— there are around 20 DSPs in the UK which currently use Athens access control;

— they are usually equipped with good technical capacity;

— they are already acquainted with the problem of authentication for external users.

The more important goal, therefore, is to minimise the cost to institutions of introducing a PKI, both in administrative terms and in demands on technical resource. The main factor here is simplicity: the scheme must be simple to implement and maintain, and should be based on each institution's existing procedures for user registration.

Simplicity is also a critical factor for the institution's users. There is little to be said for an authentication scheme which causes its end-users difficulties; unless end-users are given trouble-free procedures for using the authentication service, they are likely to place an additional burden on the institution's support staff.

6.3 Standards-based

The use of PKI technology has grown steadily since its introduction in the early 1990s, and is now a major industry-standard technology for authentication. With technology of this complexity, however, there are numerous opportunities for improvising variations on the theme, and the literature shows that there has been no shortage of attempts to do just that. One area of continuing interest is revocation which is often seen as costly and complex. Hence the development of schemes which make revocation redundant, such as those involving the use of certificates with short lifetimes.

While new ideas do emerge from time to time, few achieve general acceptance. The approach in TIES, therefore, is conservative, and remains close to base X.509 standard. This is also an important factor in enabling the integration of the ac-PKI with mainstream international PKI developments where the same conservative choices may have been made.

While there is no standard or reference implementation for PKI software, Apache web server technology does have a central role in the provision of JISC IE services. The obvious choice for implementation of the TIES software components is the mod_ssl package, which provides strong authentication for Apache based on the use of the OpenSSL open source library.

6.4 Application in FE

An original objective of TIES was to consider the deployment of digital certificates in both HE and FE institutions, and Associate Partners in both sectors were consulted. In discussions with FE colleagues, however, it became clear that significant obstacles currently face deployment in FE:

— In many institutions, personal computer accounts are not issued to students. Hence there is no existing computer identity or authentication scheme on which to build a PKI.

— Even in institutions which do issue personal accounts, resource constraints including network bandwidth limits may preclude the use of roaming profiles. In effect, a user is unable to store his/her certificate in the personal profile (held on a local filestore). So, even where a registration scheme is in place, it may not be possible to permanently install personal digital certificates on campus equipment.

Deployment of ac-PKI personal certificates across FE, therefore, will remain impractical as long as the funding for computing resources is held at present levels (it does remain an eventual objective, however). Meantime, the deployment of the ac-PKI to all HE institutions should proceed. One consequence of this finding is that the H&FE community may require a mixed economy of authentication methods beyond the period envisaged purely for migration. In effect, this may mean little change to existing practice in FE:

— IP address checking is still the predominant method for authentication to remote services for FE users. The continuation of this approach should raise no new problems.

— A secondary method, the publication within the institution of a name and password supplied by the vendor that enables access to a specific remote resource, while highly undesirable, is accepted practice for some publishers, and again raises no new problems.

This second method could be improved on, however, where an institutional certificate and key is installed as part of the default machine image loaded into all lab machines; the key would be marked as not for export. All FE lab users would then automatically present the same institutional certificate when accessing a given publisher's site. For the relying party, this has the advantage that the same PKI mechanism is used by both HE and FE users.

6.5 User confidentiality

It is a stated requirement for JISC IE authentication that the identity of the requesting user should remain confidential. The simplest way of ensuring this is to assign a pseudonym to the user, which can be traced to the actual user only by the institution responsible for its registration. As well as addressing traditional Library concerns over the confidentiality of a user's scholarly activity, this is also relevant to wider responsibilities under Data Protection legislation. Any concerns over the conduct of a specific user must be referred to the responsible institution for identification of the person denoted by the pseudonym.

In practice, the use of a pseudonym makes little real difference: an institution with several 'Malcolm Brown's must in any case rely on additional information, such as an appended serial number, to distinguish them.

6.6 Service environments

The ac-PKI will be successful only if it can be properly integrated with the existing service environments of users, DSPs, and institutions.

a) *Users*. This requirement is for proper certificate handling by mainstream browsers on the principal service platforms (Windows, Unix/Linux, and Mac OS).

b) *Institutions*. This requirement is for a practical means of user registration based on the institution's existing administrative procedures for staff and students.

c) *DSPs*. This requirement is for the validation by the DSP of the certificate presented by a user's browser. There are three cases:

   — services hosted on Unix Apache web server (used by most JISC-funded DSPs, and many local institutional services);

   — services hosted on Microsoft IIS servers (used by some institutional services);

— services provided by commercial DSPs which may use other technologies (though most are thought to use Apache).

Technical solutions must be in place in all service environments before the ac-PKI can operate effectively.

7 Stakeholders

Service issues may be considered from the perspectives of the various stakeholders in the ac-PKI:

— Users

— Institutions

— DSPs

7.1 Users

The end-users of the ac-PKI are the staff and students of H&FE institutions in the UK. Users tend to judge authentication schemes by their convenience more than anything else. For routine use of web-based services, digital certificates provide a relatively straightforward authentication mechanism. Once installed on the browser, use of the certificate requires minimal further user intervention.

Browsers vary in the methods required to install and configure certificates. There are three activities which directly concern the user:

a) *Enrolment*. The process of acquiring a certificate for installation on a browser may have a variety of technical realisations, but all require users to follow fairly similar scripts. The procedures are relatively straightforward, but some users will inevitably require assistance from their local user support services.

b) *Mobility*. Many users have access to more than one machine and need a mechanism to install their certificate and private key in two or more places. Further mobility requirements apply to the user visiting another institution or web cafe where temporary installation of the certificate and key is needed. Broadly, there are three approaches:

   (i) Permit an enrolment method which allows the same certificate and private key to be fetched from a central respository repeatedly. This has the advantage that the same method is used each time.

   (ii) Use a method that allows one-time enrolment, and instruct the user on how to create a copy of the certificate and key for physical or electronic transportation. This requires the user to follow a different procedure for second and subsequent fetches.

   (iii) Enable the user to fetch a new short-lifetime certificate as required. This removes the need for revocation of personal certificates, but precludes the method of assigning privilege (authorisation) to a user by binding it to a user's persistent identity certificate.

c) *Recovery*. It is safe to predict that users will make mistakes in handling certificates. Passwords will be lost, both those used to obtain the certificate, and those used to protect a stored certificate and private key on the browser's local key store. In general, the recovery method should be based on the existing method provided by the institution for username/password recovery, and a rerun of the standard enrolment procedure.

The main general issue may be to educate users in the responsible handling of their certificate and private key. A more certain solution may be to provide better keystore management tools than those provided as standard with mainstream browsers.

7.2 Institutions

For institutions, the main requirement is that the operation of an ac-PKI should not impose an unacceptable administrative or technical burden. At the same time, the institution is expected to exercise appropriate care in the management of registration data, in line with its approach for handling the assignment of credentials for local services. The goal, therefore, is to devise a registration mechanism that is commensurate with existing arrangements, and is not susceptible to additional vulnerabilities.

The obvious solution is to build upon each institution's existing arrangement for the assignment of credentials with the minimal extensions necessary to support ac-PKI operation. The existing model for assignment of access privilege for local services coincides with the requirements of the ac-PKI:

— The user is issued local credentials on matriculation, or start of employment.

— The user's credentials are withdrawn at the end of the course of study, or end of employment, or as a consequence of disciplinary action.

These events also form the basis of the registration activities required to support the operation of the ac-PKI. Additional procedures may be needed to cover key loss or compromise, but the principal events for granting and revoking credentials are the same in both cases.

Institutions may have some further concerns:

a) *Liability*. Institutions already undertake to follow best practice in the assignment to their members of credentials used to access licensed resources in the JISC IE. While accepting these responsibilities, institutions are unlikely to be sanguine about accepting new liabilities arising from their misuse.

b) *Enduring solution*. Institutions accept the inevitability of change, but have a proper scepticism about the adoption of a new national scheme unless there is good assurance that this is likely to endure.

On the other hand, deploying the ac-PKI should also offer a significant benefit to institutions. Instead of creating another ad hoc authentication solution for each secure local web service it wished to launch, from student accommodation booking to exam result announcement, the existence of an ac-PKI would allow the institution to exploit the X.509 client authentication mechanisms already built-in to standard web server software. There is much to be gained here, since in many institutions such web-based services are only now starting to become a reality, and alternative solutions such as on-campus single-signon, are not yet widely deployed.

7.3 DSPs

7.3.1 JISC IE DSPs

Different types of DSP provide services in the JISC Information Environment: rights-holding publishers, third party aggregators, and JISC-funded data centres. A significant difference between the JISC-funded and commercial DSPs is that the client set for JISC-funded DSPs is limited to members of UK H&FE institutions; commercial DSPs serve a wider range of clients, who may present certificates obtained from other CAs.

Several issues are common to all providers:

a) The ac-PKI should provide a dependable mechanism that enables DSPs to authenticate users requesting access to licensed resources. The DSP must be able to rely on the PKI's integrity and reliability.

b) DSPs must be confident that institutions will find administration of the scheme practical, and that it will be undertaken conscientiously. The ac-PKI Certificate Policy and Certification Practices Statement should attempt to satisfy these concerns.

c) Commercial DSPs have accepted the need to implement support for Athens as a necessary overhead in supplying the UK market. The attractiveness of a PKI solution, therefore, lies in the expectation that it will be widely applicable to commercial and academic customers, both at home and abroad. The ac-PKI should not embody non-standard features that require special handling and raise the cost of the certificate validation procedure.

d) DSPs share with institutions the concern that a new authentication scheme should be enduring, and should have a predictable implementation and maintenance cost.

7.3.2 Other resource service providers

The starting point for this report was the use of digital certificates for authentication to services in the JISC IE, in local institutions, and for international collaboration.

It may be appropriate to broaden this scope to include other services that provide resources for the academic user, notably those in the *Common Information Environment* (CIE), such as museums, national libraries, and the NHS. The immediate problem is that users of these services do not necessarily have the institutional affiliation that

would qualify them to receive ac-PKI credentials; any credentials provided to walk-in users would not be widely acceptable. Equally, some CIE services, such as those in the NHS, may be properly cautious in loosening control over their existing authentication regimes. Where it is impractical to achieve full integration between the ac-PKI and some other CIE PKI, the possible establishment of trust relationships between the two should nevertheless be investigated.

8 Risk analysis

When considering the risks to the integrity and reliability of a PKI the chief factors are the assets to be protected, the threats to which they may be exposed, the areas vulnerable to attack, and the consequent level of risk to be accepted. There is a trade-off between the degree of assurance provided and practical considerations, such as cost, and the extent to which institutions are capable of implementing the associated procedures. Certification practices which institutions find onerous to implement will not be followed conscientiously.

The main concern of this risk analysis is the JISC IE. Possible issues for the use of the ac-PKI with institutional applications and the Grid are considered briefly.

8.1 Assets

The primary assets of concern in the JISC IE are the licensed information resources themselves. All of these resources are of considerable value to their owners, since they constitute their commercial stock-in-trade. The value of these assets may be measured from the viewpoints of supplier (market value) and consumer (loss to the institution):

— *Market value*. This varies according to the resource. The majority are essentially scholarly resources, of little value outside the academic sphere. A minority, however, have a substantial commercial value in the wider world (e.g., Ordnance Survey mapping data).

— *Impact of loss*. A serious security violation could, in theory, result in the suspension of an institution's licence to access a resource, which would disrupt the scholarly activities of the members of the institution. An institution may also regard loss of reputation as a serious concern.

The CA and RA components of the ac-PKI are themselves assets whose integrity is of central importance.

8.2 Threats

Threats to the JISC IE include those which apply to the licensed resources, and those which apply to the ac-PKI itself.

8.2.1 JISC IE threats

The main threats to which JISC IE assets may be exposed are:

— *Disclosure of information to unauthorised users*. For the licensor, unauthorised access to one of the licensed resources that constitutes the JISC IE results in potential loss of revenue. This may occur through loss or disclosure of credentials, or by wilful misuse of information by an authorised user.

— *Loss of reputation*. For the licensee, unauthorised use of credentials issued to the institution constitutes a breach of the licence conditions, and may damage the ability of its members to pursue their normal scholarly activities.

For the majority of services in the JISC IE, any authenticated member of a licensed institution is authorised to access the resource. Only a few services employ an additional authorisation procedure. Hence most threats to JISC IE security arise from threats to its authentication mechanism. These include the following:

— *Unattributable assignment of credentials*. This has been a weakness of the JISC IE for some time. In certain cases, institutions permit their users to acquire credentials without attribution; in others, publishers issue a single name/password pair for use throughout the whole institution.

— *Retention of credentials*. Some institutions assign credentials with expiry times of many years (certainly more than the duration of a degree course). Others lack procedures to revoke the credentials of staff or students who leave the institution.

Other general threats also apply:

— *Disclosure of user activity*. It is a requirement of the JISC IE that a user's scholarly activities should remain confidential, subject to the institution's responsibility to ensure that its computing regulations and licence obligations are observed.

— *Absence of a Security Policy*. In the absence of a Security Policy for the JISC IE, the roles and responsibilities of the various component parts of the JISC IE remain unclear. Proper operation of JISC IE security cannot be determined without a yardstick against which to measure it.

8.2.2 Threats to the ac-PKI

Various threats apply to the ac-PKI:

— *Compromise of a CA's private key*. Security of a CA's private key is a fundamental requirement. Compromise of this key would subvert communication involving any user certified by the CA.

— *Compromise of an RA's private key*. Security of an RA's private key used to communicate with the CA is also a basic requirement. An agent masquerading as an RA could induce a CA to produce a functional, but invalid certificate. Compromise of this key would subvert communication involving any user registered by the RA.

— *Compromise of a user's private key*. The private key may be secured in various ways, typically by storage on some part of the user's filespace where it is passphrase protected. The threat is similar to the loss of a passphrase in other contexts. The consequences of this compromise are limited to communications involving that user.

— *Forgery of a certificate.* Strong authentication protects against the forgery of a certificate, provided that the CA's private key remains secret.

— *Cryptographic attack*. Progress in the development of cryptographic attacks will, over time, lead to the need for greater key lengths.

— *Data interception*. The observation by an unauthorised user of data in transit is not protected against by client authentication; an encryption service is required to counter this threat.

Clearly, compromise of a national CA is far more serious than compromise of an institutional CA. To recover from the former, new keys and certificates would have to be issued to all subjects certified by the CA, including institutional CAs, RAs, and end users. To recover from the latter, the institution would have to issue new keys and certificates to its members.

Other common threats should not arise:

— *Identity interception*. On networks where passwords are transmitted in clear, an attacker may intercept credentials by observing traffic. Strong authentication protects against this.

— *Repudiation*. The ac-PKI is not intended to provide the level of security necessary for a non-repudiation service.

8.3 Vulnerabilities

A variety of vulnerabilities may arise in the communication between components of the ac-PKI and in their operational practices. These may include the following:

a) *Physical security*. Including secure storage of private keys (including user, RA, and CA keys); access to principal system components; inadequate or insecure backups.

b) *Personnel, management, administrative*. Including availability of key personnel; insufficient training; unclear separation of personnel responsibilities; inadequate authorisation assignment as personnel change; inadequate security policies; lack of visible security commitment by management; poorly enforced security procedures; insufficient contingency planning.

c) *Communications*. Including connection of machines handling different security levels; exposure to hackers; insufficient audit trails; insufficient review of audit logs; inadequate firewall protection; inadequate virus protection; use of weak passwords.

Institutional practice varies in the extent to which these vulnerabilities are recognised, and in the steps taken to counter them. They must be considered when drawing up the CP/CPS for the ac-PKI.

8.4 Institutional applications

The ac-PKI may be used for authentication to any institutional web-based application. Institutions are free to choose on a case-by-case basis those applications they enable for ac-PKI authenticated access according to local requirements. In general, the ac-PKI provides at least the level of security as that offered by existing password-based methods.

MIS departments may be less inclined to adopt a national PKI-based solution for authentication to institutional business-critical applications before it has proved its trustworthiness (in some cases, virtual private networks are used to ensure security). On the other hand, the growth in web-based institutional applications, such as institutional portals, has created growing demand for unified authentication solutions which do not present users with multiple password challenges.

8.5 Grid applications

Each Grid resource administrator may conduct a risk analysis according to the characteristics of the resource, and clearly many Grid resources represent high value assets with correspondingly high risk factors. The acceptability of a digital certificate for Grid applications depends on two conditions:

— whether the CP/CPS adopted by the PKI promises an adequate level of assurance;

— whether the resource administrator trusts the PKI to implement its policy and practices in a diligent manner.

Many of the CAs currently serving the requirements of the Grid community provide medium level assurance (as defined by the Global Grid Forum (GGF) Certificate Policy Model [3]). The ac-PKI may provide only a basic assurance level (though the mapping between the ac-PKI and the GGF model is not precise). Consequently, basic ac-PKI certificates may be deemed at present to offer an unacceptable level of risk for Grid applications. However, if ac-PKI becomes widely accepted for use with the JISC IE and cross-institutional authentication, sufficient confidence may be built such that some Grid resource administrators may be prepared to accept basic assurance ac-PKI certification for some Grid resources.

8.6 Risk assessment

In general, the commercial owners of JISC IE assets appear fairly sanguine about the risks to which their resources are exposed, and seem content with the level of security provided by Athens access management. While there are some exceptions which require an additional registration procedure, there are no longer thought to be any which require users to physically present application forms to their librarians. Where additional registration procedures exist, these are less concerned with adding a further layer of security, and more with other purposes, such as obtaining an email address to be used for sending service announcements.

However, future services may emerge which require a higher degree of security than is thought necessary for existing services. Moreover, while the owner of a JISC IE resource may be satisfied with the present arrangements for the JISC IE, this is no argument for accepting that level of security into the future. In addition, as one goal of the ac-PKI is to develop an infrastructure capable of supporting additional local and non-local services (VLEs, Grid), the objective should be to provide the highest practical and cost-effective assurance level possible.

Overall, the most pervasive risk to the JISC IE is the lack of a Security Policy covering all aspects of its operation, not just those concerned with its authentication mechanisms.

8.7 Key theft and disclosure

The technical elegance of public key cryptography can lead to a false sense of security, particularly if the limitations arising from human factors are overlooked.

The most obvious of these in the systems considered here, is that keys are stored in ordinary PCs protected only by passwords. The actual level of security offered shares some of the limitations of conventional username/

password systems and introduces others: where physical security is weak and an open PC is left unattended for any length of time, an attacker may be able to export the victim's keys by e-mail or to a floppy disk. Browsers provide some protection against key export but this is a configurable option that users need not enable.

In some niche applications, additional protection can be provided in hardware: tamper-proof smartcards, some in the form of USB devices, and stronger user authentication through fingerprint readers or retina scanners. Such devices are likely to become increasingly widespread (indeed, during the course of this project, a portable PC obtained by one colleague came with fingerprint recognition built-in). While this kind of hardware is unlikely to be widely deployed in higher education within the short to medium term, once available it may become an important means of improving the security of private keys.

Perhaps the most common form of abuse of password-based systems which software-based PKI cannot protect against is voluntary copying of credentials, where an accredited user passes on their right to access some service to another user who has no such right. In general, a user may export unlimited copies of their private key and certificate from their own local machine.

In both cases, the remedy depends on modifying user behaviour:

— Students are already discouraged from disclosing credentials by the strictures of institutional computing regulations and the consequences of non-observance. As well as the stick, the carrot may also be used, by requiring certificate access for services the user regards as important or personally sensitive.

— Explicit advice to both staff and students on the proper handling and physical security of their private keys is a necessary element of PKI deployment.

9 Estimates

To scope the problem of establishing an effective ac-PKI, it is necessary to have some approximate estimates of key factors.

a) *Number of CAs.* Ideally, the answer would be one, but in practice, some institutions may choose to operate their own CA. The main complication for DSPs dealing with multiple CAs is the need to evaluate multiple CP/CPS documents and to interact with multiple revocation regimes with variable quality of service. With a single CA the problem is greatly simplified.

b) *Number of RAs.* Roughly, this corresponds to the number of HE institutions, which is more than 400. In institutions where registration is devolved to departments or schools, the total could be an order of magnitude greater.

c) *Number of relying parties.* At present around 20 DSPs use Athens access management to control access to around 160 services (though not all these services are part of the JISC IE).

d) *Number of certificates in currency*. The number of staff and students in UK HE institutions in the year 2000 was around two million, and student numbers have continued to rise since.

e) *Percentage of certificates revoked.* Typically, within the academic year, computer accounts are suspended only in exceptional circumstances following disciplinary action. The common approach is to renew credentials annually, or allow them to lapse. For digital certificates, there may be further circumstances in which revocation is indicated, such as suspected key compromise, but the overall requirement for revocation is thought to be modest, typically less than 0.1% per year.

f) *Number of certificate verifications.* For DSPs and certificate-using local web-based services, the combined figure is probably rather less than one verification per second.

For commercial DSPs a further question concerns the use of other PKI regimes by customers outwith UK H&FE. There are some signs of interest in the establishment of a national bridge framework for the integration of disparate PKIs in the UK (similar to the Federal Bridge CA in the US), but no firm announcements have been made. Without a means of establishing verifiable certification paths, the only option for a DSP may be to reach bilateral agreement with each closed PKI that subscribes to its services, which is practical for only a modest number of DSPs and PKIs.

10 Authorisation

The main focus of TIES is the development of a PKI for authentication in the H&FE community; other projects within the AAA programme have investigated authorisation technologies. Since most authorisation schemes are either entirely dependent on the existence of a PKI, or are, at least, capable of interworking with one, PKI deployment appears a positive enabling first step. Further, most of the licensed resources in the JISC IE are made available to all members of a licensed institution; only exceptionally is a user required to complete an additional authorisation stage (such as completion of an application form).

New authorisation solutions are necessary to satisfy both the present simple requirements of the current JISC IE, and the more complex requirements that may arise from new forms of collaborative interworking. TIES has implemented three authorisation methods to test their compatibility with PKI:

a) *Shibboleth.* The goal of this project is to create cross-institutional authentication and authorization services on the web. Put simply, it enables individuals or groups from various institutions to share resources using the credentials supplied by their home institutions (including PKI credentials); user attributes are transported from the home institution to the target where the authorisation decision is made. Shibboleth is being developed under the Internet2 initiative, and has already attracted wide interest in both the academic and commercial communities. TIES has implemented and evaluated the DSP (server) component of the Shibboleth authorisation scheme.

b) *Athens migration.* The present technology for authentication and authorisation in the JISC IE is provided by the Athens Access Management Service which is expected to operate at least until July 2006. Any successor system will of necessity co-exist with Athens for the duration of an agreed migration period. TIES has demonstrated a method for migration which combines the use of identity certificates for authentication with the use of the existing Athens service for authorisation.

c) *Licensed authority tables*. This approach was described in detail in an earlier paper [11]. The central idea is that each licensor in the JISC IE deposits information with a central authority that enumerates which institutions are licensed to access which resources. The DSP can determine from the digital certificate the identity of the requesting user's institution, and by simple lookup, can determine whether the user is licensed to access the resource. Where individual registration is also required this can be satisfied by use of ACLs.

The findings of the investigation of these technologies are presented in A.7. Use of PKI authentication with Athens and with licensing authority tables was demonstrated successfully; PKI integration with Shibboleth has been shown elsewhere by other contributors.

These technologies are of immediate interest as early authorisation solutions, but in the longer term they are likely to be succeeded by other approaches such as those based on X.509 privilege management.

11 Technical options

There are a number of areas where different technical solutions are available. These include:

— key generation,

— revocation,

— certificate profile.

11.1 Key generation

The process of enrolment, by which a user acquires an identity certificate from a CA, involves a number of steps:

— the user is registered with the CA;

— a key pair is generated (the private and public keys);

— the user issues a certification request and authenticates himself (face-to-face, or by use of a shared secret conveyed out-of-band);

— the CA binds the public key to the name of the user (the certificate subject) by generating and signing a certificate containing both these values;

— the client's browser stores the certificate and associated private key in some form of local trusted storage.

The different ways in which key generation can be performed have implications for the operation of the PKI. The main options are client-side key generation or server-side (CA) key generation. In both cases, the client and CA must possess a pre-arranged shared secret which they use to establish secure communication and user authentication. The two approaches to key generation are outlined below.

11.1.1 Client-side key generation

This approach takes a variety of forms, notably the basic scheme supported by CMP (RFC 2510) and the simple CMC scheme (RFC 2797). In both cases, the client generates the key pair and sends a certification request, including the public key, to the CA preferably over a secure connection. The CA may perform additional checks to verify the authenticity of the requestor, and then generates the certificate which is transmitted to the client, directly or indirectly.

This is the preferred approach in many PKIs, as client-side key generation does not involve disclosure of the client's private key. This is essential in higher-assurance environments, such as one where a non-repudiation service is required: clearly, non-repudiation cannot apply if the private key is known to another entity.

With client-side key generation, the user normally obtains the certificate from the CA only once (using a one-time shared secret). If the certificate and private key are to be used on another computer they must be transferred from the client by other electronic or physical means. A further feature of client-side key generation is that key recovery cannot be provided by the CA: if the user loses the passphrase used to protect the private key in the local keystore, and no backup copy is available, then the key is lost for good. (This is by no means a disaster – simply the inconvenience to the user of repeating the enrolment procedure.) The only remedy in these circumstances is to request a new certificate, which in some PKIs would cause the CA to revoke the old certificate. Given the observable frequency with which users lose passwords, it can be predicted that the majority of certificate revocation requests would arise from the routine replacement of existing credentials rather than from theft of the private key or its inadvertent disclosure to a third party. For the ac-PKI, therefore, where client-side generation is used, revocation should not result from certificate replacement following key loss.

11.1.2 Server-side (CA) key generation

Under this approach, key generation is managed centrally by the CA. The user establishes a secure link to the CA using a pre-arranged shared secret. The CA sends the certificate and private key to the user over the secure link.

In server-side operation, it is relatively simple to provide the user with a roaming server capability, which enables the user to fetch the same certificate and private key more than once. So a similar mechanism to that used initially to obtain the certificate and key may be used subsequently to recover these following loss, or to install the certificate on another computer. This may be used to install the certificate and key in a web cafe PC, or during a visit to another university.

In TIES, the roaming server was implemented as part of the CA, but may more logically be provided by a separate server.

11.1.3 Key generation in TIES

TIES has implemented server-side key generation for the following reasons:

— It has some advantages as a bulk loading method. Certificates may be generated (signed) off-line, which reduces the load on the CA at those times in the academic year when certification requests are received most frequently.

— Recovery or resetting of passwords used to access local services is a routine task familiar to any lab supervisor. For digital certificates, users may be advised to protect the key store used by their browser by setting a further password. Loss of this may occur less frequently, but is still likely to arise with some regularity. Server-side key generation simplifies the procedure for the user to recover the certificate and private key following key loss.

— Users need to use only a single mechanism to install their certificate on each of the computers they use.

These arguments are not altogether compelling, and other factors (such as a proper separation of function for key generation and certification) may tip the scales in favour of client-side key generation. The establishment of a roaming server, however, should be easier to arrange with server-side key generation and may be regarded as a decisive factor.

11.1.4 Number of key pairs

A single certificate is not sufficient for all security applications. In particular, the characteristics and management requirements of the key pair required for digital signature differ from those of the key pair required for encryption, and so two key pairs are needed if both applications are to be properly supported.

Given that the initial scope for the ac-PKI is for authentication only, a single key pair is sufficient, providing digital signature capability. It may be anticipated that demand for additional services, such as email content confidentiality, will in time lead to the requirement for the use of a second key pair for encryption, and this development should be kept in mind. Meantime, the operation of the ac-PKI is considerably simplified by the use of the single key pair model. In particular, it is important to note that digital signature certificates unlike encryption certificates have no requirements for guaranteed key recovery.

11.2 Revocation

A presented certificate may be cryptographically valid (i.e., correctly signed by a trusted CA), but still unacceptable because it has been revoked since it was issued. For example, the issuer will revoke any certificate that it discovers has been fraudulently obtained. It is not possible to tell by inspection of the certificate itself whether it has been revoked (unlike a hardware token, a digital certificate once issued cannot be physically recalled). Each time a certificate is presented, therefore, the relying party should check its validity with the issuing CA before accepting it.

For many, the issue of revocation is seen as the Achilles' Heel of PKI. One widely quoted MITRE study suggests that the distribution of revocation information has the potential to be the most costly aspect of running a large-scale PKI [6]. It is important, therefore to consider the probable costs of checking revocation status in the ac-PKI.

Two methods have been investigated: the certificate revocation list (CRL) and the online certificate status protocol (OCSP).

11.2.1 Certificate Revocation List (CRL)

This is the basic method of certificate status checking supported by X.509 since 1988, and is therefore a relatively mature solution, though still not fully supported. In this method, the CA publishes, in a standard format, a list of the serial numbers of any unexpired certificates it has issued that have since been revoked. The certificates themselves may contain the address of this list (a URL in the simplest case), known as a CRL distribution point, so that relying parties can readily find it.

Updated lists are published regularly by the CA, with the list itself containing the time the next update is due, so that the relying party can schedule the time at which it should next retrieve the list. There are some drawbacks with the CRL approach:

a) A revoked certificate will still appear valid to relying parties until the next time the list is published. This interval of vulnerability can be reduced by increasing the frequency with which the list is published, at a corresponding cost in processing overhead at both ends. Given the risk assessment for JISC information environment resources, daily or even weekly updates should be adequate.

b) Where it is impractical to hold all relevant CRLs locally (for reasons of limited storage or processing capacity, or where certificates from a large number of CAs may be presented), a relying party may opt to fetch each CRL on demand, and perform real-time checking. If the CRL is long, this could require lengthy processing which would delay user response. If the CRL is temporarily unavailable, the relying party must make an arbitrary choice between accepting or rejecting the certificate.

c) Where a relying party accepts certificates issued by a large number of major CAs and processes complete CRLs as a background task, the number of sources to consult and volume of CRL data to be processed may consume excessive resources.

Fortunately, X.509 supports mechanisms that allow these problems to be contained:

d) CRL size may be controlled by assigning subsets of all certificates issued by a CA to different CRLs; each certificate is associated with the distinct distribution point for its subset.

e) CRL processing time may be reduced by the use of delta CRLs, which contain entries for only those certificates revoked since the previous base CRL.

In general, the fewer revocations issued, the better. Provided that certificate revocation is restricted to cases of key compromise, and excludes those caused by spurious recertification requests, the numbers are readily manageable. Certificates for students leaving courses early would simply be left to expire naturally rather than being revoked.

11.2.2 Online Certificate Status Protocol (OCSP)

OCSP [9] was developed as an alternative to CRLs for use in certain PKI applications, and is a relatively new technology. Indeed, rudimentary OCSP support has only recently (in December 2002) been added to the OpenSSL open source library.

The OCSP concept is to give the relying party a real-time indication of whether a certificate has been revoked since it was issued. In the simplest case, the issuing authority maintains an OCSP responder, which can answer queries about certificates issued by that authority (those certificates will contain the URL of the responder). This immediate online response solves some of the problems of CRLs but introduces others:

a) If the relying party (the DSP) cannot establish communication with the OCSP responder within a reasonable time, either because the responder is temporarily down or due to the vagaries of the Internet, it can either refuse the certificate, even though it is likely to be valid, or take the risk of accepting it knowing that it may have been revoked.

b) An institution that intends to run its own CA should consider deploying an OCSP responder, with high availability and good performance. This may be beyond some smaller institutions. The end-user will perceive the DSP to be the party at fault even where the reason they are refused access to a service is because their home institution's OCSP responder (invisible to the user) is off the air. (This is a further argument for the use of a national or regional CA.)

c) Each OCSP response must be signed by the responder. As this is a relatively costly piece of real-time processing, the volume of requests and required response time may indicate the need for a relatively high performance server.

d) When presented with a certificate which requires validation of a lengthy certification path, the DSP must invoke OCSP requests for each certificate in the path, and perform its own path processing. In this case, the use of OCSP confers little, if any, advantage.

e) CRL technology is relatively mature, and has undergone a number of design iterations to address practical problems discovered in actual deployment. OCSP is much less mature and less widely deployed, and further development work may be required before it becomes a stable technology.

f) The CRL carries authoritative information on certificate status, and a relying party can obtain a clear view of the quality of service provided (including the freshness of the CRL). With OCSP, much of the information which may be relevant to the evaluation of the certificate's validity is not available to the relying party.

Note also that an OCSP service is typically driven from the set of relevant CRLs, and in this case cannot provide more timely information than that contained in the CRLs themselves.

11.2.3 Revocation requirements for the ac-PKI

A variety of strategies for checking certificate status are available to a DSP:

a) *OCSP*. This obviates the need for any CRL processing, and so appears to be the simplest option for a relying party. In effect, the cost of this processing is off-loaded to the OCSP responder (though this advantage is lost where more complex certification path processing is required). This may be an attractive solution for some DSPs, but does depend on support by the CA, with appropriate sizing to ensure the responder performs acceptably.

b) *On-demand CRL checking*. The relying party fetches and processes the relevant CRL on demand from an LDAP server [8] or web server. This is less likely to provide good performance, but may be practical where the range of issued certificates is partitioned into manageable subsets at distinct distribution points.

c) *Comprehensive background CRL processing*. The relying party downloads CRLs from its set of approved CAs according to the configured CRL update frequency, and processes these as a background task. This can produce a local storage representation that enables very efficient look-up.

For JISC-funded DSPs, option (c) appears the most economical solution. There are few CAs (preferably one), with a modest number of revocations, a modest update frequency, and no real-time response issues. Overnight CRL processing is consistent with a daily update frequency, and the use of delta CRLs containing only newly revoked certificates would further reduce the processing load. Real-time inspection is fast as the revocation data can be economically represented by an in-store bit map.

For commercial DSPs the problem is less well-contained. These DSPs may have commercial customers who use a variety of CAs with different certification and operational practices, including revocation handling. While the OCSP solution may be immediately attractive to commercial DSPs, this is not yet sufficiently widely supported to be considered a viable option.

For local institutional web-based services with simple certification path processing requirements, OCSP may be regarded as a possible niche solution which can be accommodated with minimum impact and low maintenance cost.

In principle, support for all three options would address the full range of possible DSP requirements and help reduce the cost of participation in ac-PKI. Background CRL processing is the simplest option to provide; the requirements for high availability and good response times for the OCSP and on-demand CRLs makes these options marginally more expensive, but some demand may exist for both.

Certificate revocation options are discussed further in A.9.

11.3 Certificate and CRL profiles

TIES certificates conform to the certificate profile defined in RFC 3280 [10]. Further details of the *subject*, *validity*, and *extensions* fields are given below.

11.3.1 Subject

The subject field identifies the entity (user) associated with the public-key found in the subject public key field. It comprises an X.500 distinguished name. The following two naming schemes may be operated in parallel:

a) Country name (from ISO 3166 [4])
Institution name (as Organisation, drawn from a standard controlled list)
Optionally, one or more Organisational Unit names
Personal pseudonymous identifier (as Common name)

For example:

```
C=GB
O=University of Edinburgh
CN=a8f0khHqMgK
```

b) The institution's DNS name (as Domain component names)
Optionally, one or more Organisational Unit names
Personal pseudonymous identifier (as Common name)

For example:

```
DC=uk
DC=ac
DC=ed
CN=a8f0khHqMgK
```

Note that in Microsoft Certificate Services where the subject name is generated automatically from the user's Active Directory entry, there may be constraints on the choice of subject name.

11.3.2 Validity

Every digital certificate contains a validity field which specifies the time interval during which the certificate is considered applicable. In practice, it specifies the period for which the CA guarantees it will maintain information about the status of the certificate (the CA will cease to publish revocation information about a certificate once it expires.) The choice of validity period depends on the balance of a number of competing factors. Arguments for a shorter (one year) validity period include:

- For HE institutions, the natural period in the cycle of activity is one year. The major milestone events of matriculation and graduation are annual events. As new students must in any case be registered annually with the CA, this is also an appropriate time to reregister existing users.
- The cost of CRL processing is borne by every relying party, and is, in part, proportional to the volume of revoked certificates to be handled. As the number of revoked certificates expands over time, the performance of relying parties would benefit from shorter validity periods.
- Where a user's institutional status becomes ambiguous (perhaps during a period of absence), and no specific event actually triggers a revocation request, the period of exposure during which the certificate remains useable is limited to one year.

Equally, there are counter-arguments for longer validity periods:

- Before an existing certificate expires the user must acquire a new certificate to ensure continued access to information resources which rely on user certification. The predictable user preference would be for this procedure to be required as infrequently as possible.
- From a user perspective, the typical period for which a student is affiliated to an institution is for the duration of a degree course. Hence three years is an appropriate period for certificate validity.
- Different categories of user should be assigned certificates with different validity periods. For example, a period of five years may be thought appropriate for staff members.

In broad terms, the trade-off is between user convenience and PKI performance and integrity. The latter appears a stronger case, and the advantage of the one year certificate can be summarised as follows:

- *Cost*. While it is entirely within the capability of institutions to manage registration and revocation for users assigned different validity periods of varying durations, this does entail additional management costs. At the same time, for relying parties, a one year validity period places a cap on the overheads arising from CRL processing.
- *Flexibility*. It is difficult to change certain characteristics of a PKI once deployed. With a one-year certificate life-cycle coordinated across institutions the slate is wiped clean annually, and operational changes (such as those that might arise from a change in certificate profile or in CA supplier) can be accommodated more easily.
- *Exposure*. Where an institution fails to request revocation for a user, the period during which the certificate remains useable is limited to one year.

In practice, to accommodate routine key replacement a validity period of 12+ months may be required. In a well-engineered PKI it should be possible to establish different certificate rollover procedures for different categories of user. While all users may have a basic validity period of 12+months, staff certificates may be rolled-over automatically over a five year period, while student certificates may require manual annual rollover.

11.3.3 Extensions

The following standard X.509 certificate extensions are used:

a) *Key usage*: This indicates that the certified public key is intended only for verifying digital signatures (for entity authentication). It is flagged non-critical.

b) *Certificate policies*: This identifies the certificate policies supported by the ac-PKI CA that apply to the certificate. It is flagged non-critical. This may include an identifier for the policy operated by the user's RA.

c) *Basic constraints*: This constrains the certificate subject to the role of end-entity, rather than CA. It is flagged critical.

d) *CRL distribution points*: This identifies the CRL distribution point or points to which a certificate user may refer to determine whether the certificate has been revoked. It is flagged non-critical.

The following extension is defined in RFC 3280 [10]:

e) *Authority information access*: This identifies the location of the OCSP responder to which a certificate user may refer to determine whether the certificate has been revoked. It is flagged non-critical.

There is some argument for making all extensions critical, to ensure proper operation of the PKI. On the other hand, if some certificate verifier does not support a particular extension (marked critical), it might reject a good certificate. Further, it could be argued that a certificate verifier that ignores a non-critical, but significant extension does so at its own risk. The setting proposed above is chosen as a compromise between these positions.

A number of vendors (including Netscape, Microsoft and Entrust) have defined private extensions. TIES certificates did not include any of these extensions and no problems were detected resulting from their absence. In some circumstances, however, it maybe necessary to include vendor-specific extensions to ensure the correct operation of certain functions in vendor products.

11.3.4 CRL profile

Version 2 CRLs are required. A number of CRL extensions may be relevant, including: CRL number, reason code, issuing distribution point, and delta CRL indicator.

11.4 Use of Directory

At the outset of the TIES project it was assumed that it would be necessary to establish a Directory as a repository and a means of publication for certificates and CRLs. On further examination of the requirements, a number of factors relevant to certificate/CRL publication emerged:

a) The scope of the project is limited to simple user authentication. In particular, encrypted email (which involves knowledge of a recipient's public key which may be obtained from a Directory) is not an immediate requirement.

b) The scope is also limited to resources accessed using standard web browsers. Since browsers that support client-side SSL authentication [13] do so by pushing the requesting user's certificate in the course of the security exchange, the web server does not require a mechanism to fetch this certificate from a third party.

c) CRL data is typically made accessible in a variety of ways, including simple HTTP access. In addition, other certificate revocation checking methods, such as OCSP, may be available which do not depend on Directory access.

So while it would be forward-looking for the CA to publish all issued certificates and CRLs in a public Directory, this is not an immediate service requirement.

12 PKI architecture

A PKI may take a variety of forms, determined by the trust relationships between its component parts. The simplest PKI architecture is represented by the single CA. This is straightforward to deploy and operate, but is limited by the extent to which the collection of users can be covered by a single set of technologies, policies, and methods. Separate PKIs can interoperate only if they are linked in some way, such that trust relationships can be established. The two common approaches for connecting CAs that support distinct communities are the hierarchical PKI and the mesh PKI (trust list).

In the hierarchical model, the common trust point is provided by a root CA, trusted by all subordinate CAs. These subordinate CAs can interoperate as each has a verifiable certification path back to the root (each subordinate CA has been cross-certified by the root CA). While a hierarchical PKI will scale to accommodate additional communities of users, it shares some of the drawbacks of the single CA. It too represents a single failure point (if compromised), and depends on acceptance by subordinate CAs of a common set of policy and technological constraints. In some sectors this may be difficult to realise on both practical and political grounds.

In a mesh PKI, CAs are connected in a set of peer-to-peer bi-lateral relationships (by one- or two-way cross-certification). There is no single trust point: trust is established by bi-lateral agreement between CAs. This is a highly adaptable model, capable of incorporating new communities and is relatively resilient. Independent CAs can establish trust relationships without disrupting existing relationships. Unfortunately, certificate validation becomes complicated, as the construction of a certification path is no longer deterministic. The certificates themselves become complex, as they may need to contain sets of policy mapping information to make path construction possible in each case. Note also that long certification paths carry a performance penalty for relying parties; simpler CA configurations enjoy a relative efficiency advantage.

US federal agencies encountered problems with both these models when attempting to establish a federal PKI for secure communication across the range of government organisations [1]. It was first assumed that a hierarchical model would match the inherent hierarchical nature of government organisation. But the attempt to agree the appropriate organisation to operate the root CA was unsuccessful, and was finally abandoned after individual agencies had established independent PKIs. The alternative approach of developing a federal mesh PKI was then considered, but found impractical given the complexity of the trust relationships that would be required. The solution the US has now adopted uses Bridge Certification Authority (BCA) architecture.

The BCA does not itself act as a trust point for the users of the autonomous PKIs that it bridges. Rather, it enables the establishment of peer-to-peer trust relationships between user communities by specifying a minimum level of assurance to which all CAs must conform. Hence its role is less about technology, and more about providing a mediation service that enables PKIs to reach agreement on mutually acceptable trust relationships. A demonstration of the Federal BCA (FBCA) was conducted between 1999 and 2001 which provided proof of concept, and development has continued since then. In general, while it appears that the use of a BCA as a mediation agent between PKIs is a pragmatic solution, it is a complex problem to address, and must resolve relevant policy, legal, organisational and technical issues.

Given the organisational complexity and diversity of colleges and universities in the US, arising in part from fiscal autonomy, it is not surprising that the BCA model has been proposed for US HE: the Higher Education Bridge Certification Authority (HEBCA). A prototype scheme was successfully demonstrated in 2002, involving the verification by NIH of grant applications signed by researchers in three universities. This is intended to enable trusted communication both between HE institutions, and between HE and federal agencies.

In UK HE, the situation is much simpler. The role of centralised agents for the management of infrastructure, such as the network, is well established, and there is no major legacy of existing PKIs providing client authentication. Consequently, the simple model of a single CA (or, if necessary, a two level CA hierarchy) should be adequate for the job. However, in the expectation that the ac-PKI will connect to other national and international PKIs, the ac-PKI should consider interworking issues when formulating its technical and policy solutions.

13 Existing institutional PKIs

Many institutions have already deployed an internal PKI on a somewhat ad hoc basis to support SSL-secured communication on local servers (server-side authentication), in some cases involving hundreds of servers. There are several ways in which these internal PKIs could be handled within the ac-PKI.

a) *No action*. These institutional PKIs are self-contained and not used for cross-institutional activity. There is little to be gained, therefore, from subsuming them within the ac-PKI. On the other hand, it can be predicted that applications requiring cross-institutional secure server communication will arise.

b) *Assign ac-PKI cross-certificates to institutional CAs.* This appears to provide a simple win, enabling cross-institutional security using existing infrastructure. However, there are some major concerns:

(i) the properties of many existing deployed certificates makes them unlikely to be suitable for cross-institutional use (for example, absence of usage constraints);

(ii) certificates issued by different institutions are likely to vary considerably in content, creating the potential for technical incompatibility;

(iii) institutions should not reuse certificates issued for local use before considering relevant policy and organisational issues.

c) *Assign ac-PKI certificates to institutional servers*. This has two main advantages:

(i) it places institutional certificate use on a more rational basis, with proper consideration for certificate properties (notably, policy-related attributes);

(ii) it provides a pervasive capability for an SSL-enabled server to engage in cross-institutional secure communication;

(iii) it is more economical to operate one PKI than two.

The main disadvantage is the possible administrative effort in satisfying the formal requirements of the ac-PKI policy for enrolment of numbers of institutional servers.

The best choice in the short-term may be a combination of options a) and c):

— maintain present arrangements for management of certificates for servers that have a purely local role;

— provide an ac-PKI registration mechanism for the enrolment of servers that institutions wish to use for cross-institutional secure communication.

In the longer-term, full integration of internal institutional PKIs with the ac-PKI may provide significant operational benefits.

14 Policy requirements

The main players in a PKI are the end-user (certificate subject), the certification authority, and the certificate user (relying party). Each entity has obligations to the other two, and in return, enjoys qualified warranties offered by them. These obligations and warranties are defined in a certificate policy and certification practices statement (CP/CPS). All certificates are issued in accordance with some policy, even if that policy is not made explicit. For a certificate to be meaningful to a range of relying parties, however, clear policy and practices statements must be developed and made publicly available.

Of course, a simple declaration of adherence to a policy does not in itself satisfy the assurance requirements of the other parties. In general, trust that the policy has been conscientiously administered grows over time, with repeated demonstration of expected behaviour. According to the application for which a certificate is to be used, a relying party will seek a commensurate level of assurance that the presented certificate is both valid and reliable:

— that the certificate was issued by a CA according to a policy providing an acceptable level of assurance;

— that the certificate was issued to a qualified individual identified by the certificate subject name;

— that the private key associated with the certified public key is in the sole possession of the certificate holder (with perhaps a copy cached by the CA).

14.1 Policy classes

The FBCA Certificate Policy Model [14] defines five policies, which represent four different assurance levels (rudimentary, basic, medium, and high) plus one for testing purposes. These classes reflect the rigour of the certification process and the corresponding degree of confidence a relying party may assume at each level. It is the responsibility of the relying party to evaluate the threats and vulnerabilities and determine the level of risk they are prepared to accept based on the sensitivity of the resource. The four policies vary in the rules applied to the following aspects of CA operation:

a) *Roles*. The number of people or roles responsible for different aspects of CA operation (indicating the degree of separation of function).

b) *Authentication of identity*. The rigour of the procedures used to verify proof of identity of the end-user before certification. Note that more explicit guidelines for the setting of trust levels for registration and authentication have been defined [7]. These may be used in conjunction with the FBCA model.

c) *Routine rekeying*. The rigour of the procedures used to verify proof of identity of the end-user when issuing a new certificate to replace one approaching expiry.

d) *Revocation time*. The time limit within which a revocation request shall be acted on.

e) *Audit details*. The types of event recorded in audit records, frequency of inspection, and archival policy.

f) *Private key storage*. Whether the private key may be stored in software, or is held only in a hardware security device.

As indicated above, the probable assurance level required for JISC IE and institutional applications is basic assurance, and for Grid applications is medium assurance. The obvious solution for a single CA serving the needs of UK HE would be to support two CPs: one for basic assurance certificates issued for JISC IE and institutional applications, and one for medium assurance certificates issued for use with the Grid.

14.2 Grid

There is no single rulebook for designing a PKI that will satisfy the requirements of Grid applications; the GGF policy model provides guidance, but leaves much room for interpretation. In practice, a CA applies to a policy board for recognition, and is subjected to examination (possibly involving a viva voce for the applicant). Once accepted, holders of certificates issued by the CA are thereby authorised to access certain general resources and to apply for accounts on other resources. So there is a high barrier for an agency to gain recognition as an accepted CA, another for a user to gain certification, and a third for a user to gain authorisation to use a specific resource.

A possible variation of this model might involve a procedural shift from the authentication function to the subsequent authorisation function:

a) A national CA should be able to gain widespread acceptance as a responsible agency.

b) An institution operating a bulk authentication scheme, based on registration information contained its master administrative databases, provides a credible level of assurance for user identity.

b) Separately administered authorisation schemes could entail additional procedures involving presentation of physical evidence, email-addresses, a letter from the Head of Department, or personal endorsement by an existing certificate holder. Each authorisation process would confer privilege on the identity certificate holder to access some specific resource.

By the use of these additional measures at authorisation time, basic assurance certificates (which actually provide reasonably satisfactory identity assurance) may become acceptable for certain Grid applications.

14.3 US HE model certificate policies

In the US, one approach supported by organisations including EDUCAUSE, Internet2, the NSF Middleware Initiative, and CREN, is for the establishment of institutional CAs, and draft policies have been produced to assist institutions in developing their own CP/CPS. For example, the lightweight (PKI-Lite) CP/CPS aims to support a relatively simple framework consistent with existing methods of campus central registration and authentication. (With the dissolution of CREN, there is currently no CA issuing institutional certificates for the PKI-Lite initiative, though a new CA is promised.)

A more substantial framework is provided by the HEPKI Model Campus Certificate Policy [2]. This is largely a reworking of the CP developed by the FBCA [14]. The choice of a BCA approach indicates the expectation that a bridging solution will be required to enable a large number of HE CAs to interoperate. Unfortunately, the use of bridging to link numerous PKI islands is not a simple matter. Problems may be anticipated in the interpretation of different institution's CPs, leading to the rather unsatisfactory reliance on policy mapping mechanisms.

The fact that the US has lacked the national HE agencies found in the UK may have given them little choice but to adopt the BCA approach. In the UK, however, the alternative model of a national H&FE CA, with institutional RAs is available, and unlikely to place much strain either on the capacity of institutions or on the capabilities of the supporting X.509 technology.

14.4 Two-tier policies

One possible approach for the provision of higher assurance certification would be for institutions to adopt a two-tier model for registration, using the simple registration mechanism described here for students, and a more conservative method for staff. The latter might involve application in person with presentation of staff card, one-time enrolment, and client-side key generation. This need only apply to staff who wish to apply for access to resources that demand higher assurance certification, such as the e-Science Grid. For the majority of staff, basic assurance credentials should be sufficient. The registration mechanism used in each case would be reflected in the certificate policies recorded in the certificate.

On the other hand, the need to employ distinct registration mechanisms may be regarded by some institutions as an unacceptable complication. Establishing two mechanism effectively doubles the cost to the institution. Moreover, if the demand for medium assurance certification remains weak, institutions may conclude that the additional cost is not justified by the demand.

It may be possible to find a compromise solution, involving a single registration procedure for all members of the institution, and a two-tier enrolment procedure operated by the CA. Each registration request issued by the institution would indicate the assurance level to be applied during enrolment (corresponding roughly to the staff/student roles). So while the same mechanism for registration applies to both staff and students, staff are required to follow a more rigorous enrolment procedure (involving client-side key generation, one-time enrolment, use of email exchange, telephone confirmation, et cetera). As staff may be expected to be more responsible than students in the care they exercise in handling credentials, recertification arising from key loss should be an infrequent occurrence. Staff certificates may also have a longer lifetime, or (better), undergo automatic key rollover.

To summarise, there are three possible approaches for simplifying the provision of appropriate assurance levels required by the users of ac-PKI certificates:

a) Relax the requirement for medium level assurance by making additional checks at authorisation time on users holding basic level assurance certificates (see 14.2).

b) Require institutions to operate a two-tier registration model (basic and medium).

c) Retain the standard registration model, and introduce a two-tier enrolment procedure for the CA.

14.5 Policies for ac-PKI

In addition to the policies for certification of end-users, further policies are required for certification of institutional servers. In total, three policies are required:

a) *Basic level user certification*. This applies to certificates used to access JISC IE services, and most local institutional services (at the discretion of the institution). It is intended for certificates assigned to students, and to staff who do not require access to resources that demand higher-level assurance. The majority of users would use this form of certificate.

b) *Medium level user certification*. This applies to certificates used to access higher-level assurance resources, such as the e-Science Grid, but may also be used by staff for accessing sensitive resources within institutions.

c) *Institutional server certification*. This applies to certificates used by institutional servers involved in cross-institutional communication. One instance of this common to all institutions is certification of the institution's RA.

Using the assurance categories defined in the FBCA policy model, item (a) provides basic level assurance; the others provide medium level assurance. The use of a single CA issuing certificates according to these three policies simplifies the task of ensuring they remain consistent in both technical and operational terms:

— it enables basic assurance certification to be carried out under more stringent operational arrangements than would be possible using multiple CAs;

— it provides an institution with a single line of communication to the authority responsible for all its certification.

The ac-PKI root CA will provide the following types of certificate:

— cross-certificates for institutional CAs (if any);

— certificates for institutional RAs;

— certificates for members of those institutions.

15 Total cost of ownership (TCO)

The concept of Total Cost of Ownership has emerged as a business tool to assist organisations in evaluating the costs/benefits of the various options for the deployment of specific IT products or services. This arose from a 1996 study which concluded that the annual maintenance costs of a Windows 95 desktop were five times the purchase price. While the task of calculating the TCO is a complex if not esoteric undertaking, with a virtually inexhaustible supply of relevant factors, the concept itself has some value when considering options for setting up the ac-PKI.

The present TCO for authentication services in the UK H&FE sector is substantial, and includes the following:

— costs of maintaining local authentication services, either stand-alone, or in a single-signon framework;

— annual Athens licence for the sector, and individual Athens licence and support costs for DSPs;

— operational costs for the UK e-Science Grid CA.

While a price can be attached to the costs of Athens and the e-Science Grid CA, the cost to institutions may be more readily estimated in staff time that could be put to better use.

There are three options for the provision of a central CA:

a) establish a CA within the academic community, managed by an existing academic body, and implemented using open source software (as with the UK e-Science CA);

b) insource an in-house PKI based on software licensed from a commercial vendor, implemented by an existing academic body.

c) outsource to a commercial vendor for a managed PKI solution;

It might be thought that these options are ordered by increasing cost, and decreasing risk of failure (with the outsourced solution the most likely to succeed, but at the highest cost). Without conducting a full TCO analysis this may be difficult to judge. Verisign have produced a TCO analysis [12] that argues the contrary, and concludes that the cost of a fully outsourced PKI is significantly less than buying the PKI software and running it in-house. There may be some question over whether this advice is offered altogether dispassionately, or is the predictable argument a vendor uses to pacify alarm over the up-front price.

Nevertheless, this holds out the attractive possibility that the most technically effective solution is also the most cost-effective, and bears further consideration. The arrangements for the establishment of the ac-PKI would have to ensure that it was always possible for the community to change supplier at the end of each contractual period.

Further investigation of the commercial options is required before final conclusions can be reached.

SECTION III – CONCLUSIONS AND RECOMMENDATIONS

16 Conclusions

The findings of the detailed technical investigation discussed in annex A have informed the conclusions presented here.

A number of issues have been considered relevant to the realisation of the ac-PKI:

- *Initial questions*
- *Effectiveness*
- *Architecture*
- *Procurement*
- *Policy*
- *Technical issues*
- *Preconceptions*

16.1 Initial questions

Four questions posed at the outset of the project may now be reconsidered:

a) *Can an institution fulfil the RA function by use of its existing registration procedures?*

This should be straightforward for most HE institutions. Typically, institutions already export staff and student user data from central HR and Registry databases to populate the user/password lists of various local servers. The requirements for a local RA are satisfied by a similar process. The RA function requires only one essential data item to be added to the basic user record: the external username assigned for use as the certificate subject name; an audit record detailing the interactions with the CA is also required.

At its simplest, the RA function could operate semi-autonomously, assigning and recording registration data (external username, validity period, passphrase, registration date, revocation date) without reporting these back to the central databases. In most cases, however, it would be sensible to integrate the RA function and the data it generates with the central user record management system.

In FE, many institutions either do not assign personal accounts, or could not easily accommodate personal certificates on campus equipment. As an interim measure, the installation on campus browsers of a single certificate for the institution could replace existing use of publisher supplied name/passwords.

b) *Can an institution implement its own CA at reasonable cost using open source or commercial components?*

An initial assumption was that a number of institutions would opt to implement their own CA based either on Unix or Microsoft software:

(i) *Unix*. OpenSSL provides a functional toolkit for cryptographic processing but does not claim to provide more than a rudimentary CA. A considerable amount of work would be required to implement a fully functional CA from these components.

(ii) *Microsoft Certificate Services*. This comes bundled with Windows 2000 Server, and provides CA functionality that interworks at protocol level with other technical components prototyped in TIES. This has been designed for the commercial enterprise market and is integrated with other major components in the Microsoft systems architecture, such as Active Directory, which is used to derive the certificate subject names. (Note that since Active Directory typically holds real names, these certificates will not usually be pseudonymous.) The cost of entry is low if Active Directory is in use, though some institutions may have concerns over single-vendor lock-in.

Given that an institutional CA can be built (in the case of Microsoft, with little apparent effort), the question remains whether it makes sense for institutions to take this route. In both cases, the solution appears rather naive, and falls well short of the functionality expected of a dedicated CA. Specific shortcomings of the open source approach include:

— No web-based user interface is provided; this must be developed and maintained locally.

— The mechanism for downloading a certificate to the browser must be developed, but is unlikely to be as functional as that provided by a commercial CA.

— Commercial CAs often provide other browser enhancements, such as plugins for improved private key management.

— No standard RA interface is available. This must also be developed from scratch.

— No standard certificate store is provided. A suitable database technology must be chosen and the database designed and implemented.

— Only rudimentary (single-threaded) OCSP responder support is provided.

— CRL handling is also rudimentary.

— Only the basic software is provided. There is no consultancy advice for developing administrative procedures to ensure CA security and availability, and essential provisions such as backup and recovery strategies.

— There is no indication that substantial issues arise in real operation, such as the separation of function in operational personnel.

A cheap-and-cheerful roll-out implementation may appear attractive, but is unlikely to produce certificates of general acceptability in the wider world.

c) *Can a service in the JISC IE implement certificate validation using open source or commercial components?*

Certificates become useful only where they provide relying parties with verifiable evidence of user identity. The question, therefore, is whether the tools exist that would enable JISC IE service providers to implement reliable certificate validation. Again, there are two web server architectures to consider: Unix/Apache and Microsoft IIS:

(i) *Unix/Apache*. Apache 1.x with mod_ssl and OpenSSL was found to provide a reasonable solution. CRL handling, however, is rudimentary and would require substantial work to make fail-safe. Commercial and JISC-supported DSPs should have little difficulty in implementing basic certificate validation for Apache (see A.5).

(ii) *Microsoft IIS*. This provides a functional solution. Weaknesses were discovered in the areas of CRL handling and audit trace, but workarounds are possible for both (see A.6).

For both architectures, additional implementation would be required to provide a more comprehensive and reliable solution. Some unknowns remain in the ability of this software to verify certificates involving more demanding requirements, such as complex certification path processing and the enforcement of policy constraints. Only basic validation was investigated within the project.

d) *Is support for certificates in the mainstream browsers for each of the common desktop platforms functional, and useable?*

The mainstream browsers were functional on all platforms (with the partial exception of Mac OS-X, for which only Mozilla could be used). In terms of usability, there are some distinctly rough edges, and institutions will need to invest some effort to provide guidance to users. A number of US universities appear to have overcome usability issues in their large-scale deployments of digital certificates. The main problem for the user is the lack of a fully engineered key management framework, capable of automating the process of installing and renewing private keys.

16.2 Effectiveness of X.509

The following considerations apply to the effectiveness of X.509 digital certificates as the basis of an authentication scheme for UK H&FE:

a) An access management service is required to ensure that only members of authorised institutions are able to access licensed resources in the JISC IE. The Athens system has served this purpose well for several years, but has a number of drawbacks: it is proprietary, it requires modification to web server software, and it is a UK-specific solution. In contrast, basic PKI technology is standardised, in widespread use, and is deployed in mainstream browsers and web servers.

b) PKI is capable of addressing many of the security requirements within H&FE: JISC IE services, the e-Science Grid, and secure institutional applications. The development of a unified ac-PKI precludes the need for the more costly approach of building a set of isolated PKIs, or maintaining a plethora of authentication schemes, each serving a distinct group of collaborative workers in UK H&FE.

c) Many of the authorisation schemes currently proposed are either wholly dependent on PKI or are, at least, capable of interworking with it. Hence the ac-PKI will provide a necessary enabling mechanism, regardless of which authorisation scheme is eventually adopted.

d) One obstacle to the deployment of the ac-PKI is the cost to institutions in terms of administrative and technical resource. This can be made more affordable by offloading the CA function, leaving institutions with the more tractable role of RA. It has been shown that standard institutional practice in the management of staff and student databases can be used directly to support the registration function. The main additional data to record is the external username assigned as the user's certificate subject name, and the audit history of data transmitted to the CA.

e) The main attraction of the ac-PKI for institutions may be its use for controlling access to local web services. Within the framework of a deployed ac-PKI, the method of protecting web pages so that only certificate-authenticated users can access them is by simple configuration (see A.5.2 and A.6.2).

   Unlike JISC IE services which enable access according to institutional affiliation, local services will more often need to identify users individually. Individual identification (the subject common name) is present in each ac-PKI certificate, which is readily available to server-side scripts. In the case of Apache/mod_ssl, a script is presented with most of the significant components of the client certificate as strings in environmental variables, including the subject common name. The service may have to map this pseudonymous subject name to a local username, but the identification of the user account is otherwise straightforward.

   Once introduced, the more local services that make use of the ac-PKI for authentication, the lower the overall cost per service will be. This should be an incentive to maximise the use of certificate authentication for local services in preference to other methods.

f) While there are considerable attractions in a PKI solution, there are also some risks: it is not yet a fully costed solution; the likelihood of uptake for local institutional and E-Science Grid use is unproven; timely deployment by commercial DSPs is not assured. Equally, a variety of other authentication and authorisation solutions are vying for attention: a number of bespoke single-signon solutions; the use of name/passwords over encrypted SSL links; the perennial Kerberos; the aspiring Shibboleth.

   Of course, each of these alternatives tends to address a different subset of the problem area; none has the scope of the PKI solution.

16.3 CA architecture

There are a variety of architectural options for the ac-PKI:

— it may be implemented using a single CA with per-institutional RAs;

— alternatively, each institution could operate its own integrated CA/RA;

— equally a hybrid solution may be used, with some institutions operating their own CAs, and others offloading this function to a national or regional CA.

Where multiple CAs are involved, they may be arranged hierarchically or in a peer relationship. The following considerations apply to the choice of CA architecture:

a) Operating an institutional CA is hard, and requires knowledge which few institutions have found necessary to acquire. (The various attempts to develop open source CA software demonstrate that this is hard even with knowledge of the subject.) While this project has shown that it is within the capabilities of most HE institutions to implement a rudimentary institutional CA using open source or packaged commercial software, this does not take account of the operational, organisational and policy issues necessary to engender trust.

b) Operating an institutional CA confers no advantage to the institution. For example, the ability to design a local schema for certificate subject names, or to add non-standard extensions may be thought beneficial for some local uses, but is likely to diminish the acceptability of these institutional certificates externally.

c) The alternative option of establishing regional CAs also appears less attractive on examination:

   — it loses the advantage of economy of scale intrinsic in a single national CA;

   — a certificate verifier may be less inclined to trust a set of regional CAs than a central CA;

   — a regional CA is likely to operate at a similar assurance level to that provided by an institutional CA.

d) The Grid provides an example of interoperation of independent PKIs linked to one another in a mesh relationship. The experience of dealing with multiple CAs has caused the GGF to move towards the model of single national CAs. In general, unless genuine differences in common interests exist which require multiple CAs with multiple trust relationships, it is preferable to minimise the number of CAs in the PKI.

e) It is substantially more difficult to build an integrated PKI from a collection of disparate PKIs than it is to construct one from scratch (consider the need in the US for bridge CAs). Given the lack of well-established institutional PKIs in the UK, it makes good sense to put in place a coherent national solution while this is still a green-field development. Crucially, it ensures that H&FE will adopt a single set of CP/CPS, rather than a range of statements which relying parties would need to consider separately.

f) Deployment of a single CA simplifies the establishment of trust relationships with other European and US PKIs. Thereby, common trust relationships may be widely established between users and services in the UK academic community and those in European and US PKIs.

16.4 Procurement

There are three main choices for procurement of a CA for the ac-PKI:

— develop within the community using open source software;

— develop in-house using PKI software purchased from a vendor;

— outsource entirely to a vendor.

Recent discussions at the Educause PKI Workshop, Snowmass, August 2003, reported by D.W.Chadwick, have a bearing on this. It was announced that the US Federal government have found that outsourcing a CA is always significantly cheaper than running your own CA in-house. The US has stated that government agencies will no longer be allowed to implement an in-house PKI. They must now buy in the PK infrastructure from a commercial vendor and use the government root CA as their trust anchor.

The government also found that the cost of a biometric hardware token for storing a user's private key is less than the cost of password management of tokens without biometrics. The cost of PIN/password management only reduces significantly *when users have to use their private key password every day*. Having a tick box, "remember password" (as in IE for example), defeats this, as users no longer have to remember their passwords. A solution that eliminates the support costs associated with forgotten passwords is more economical over the lifetime of the PKI. While these observations may have arisen from deployment at a different level of complexity and expenditure to that envisaged for the ac-PKI, they are relevant findings based on actual experience.

It was also stated that the RA is a major cost centre because of the staff costs in employing people to administer user registration. Minimising this time is a key component of reducing overall costs. (The proposal to bind the registration function to the existing administrative procedures for user induction therefore appears sensible.)

TIES found that the development of a CA using open source software did not provide a satisfactory solution. The ac-PKI will require a highly professional facility that promotes management support, rigour, robustness, and high availability. No firm evidence was available on the relative merits of insourcing and outsourcing, and further work is required to investigate this question further.

The initial cost of implementation of a PKI is high, but should diminish over time as further applications make use of it, and the existing costs associated with authentication for those applications are eliminated. For the ac-PKI, therefore:

— there is an initial speculative investment;

— to achieve cost-effectiveness, use of the PKI must extend to other institutional applications requiring routine presentation of the user's certificate.

This may be more readily achievable as institutions come to realise the value of the ac-PKI certificate as a means of single signon for all their local web-based services.

16.5 Policy

Policy determines the technology, management, organisation and implementation of the PKI. Unless the policy reflects a good understanding of the functions, level of trustworthiness, and responsibilities of the parties involved (users, RA operators, CA operators, DSPs), and the risks to which the protected resources are exposed, the PKI is unlikely to serve its purpose.

Several aspects of policy have been considered:

a) *Policy models*. Two very different policy models have been proposed for HE in the US:

   — the PKI-Lite model proposes a simple two-level hierarchy of institutional CAs with a single trust anchor;

   — the HEPKI model proposes a bridge solution, linking heterogeneous collections of institutional and federated CAs.

   In the UK, the considerably simpler model of a national ac-PKI would satisfy current requirements, and, given the absence of established PKIs, will not encounter legacy issues.

b) *Assurance level*. To cover the full range of applications for the ac-PKI, it should support both basic and medium level assurance (as defined by the FBCA). The following forms of certification are proposed:

   — *Basic level user certification*. Used to access JISC IE services and most local institutional services.

   — *Medium level assurance certification*. Used to access higher-value resources, both on and off campus (each institutional RA will require medium level assurance).

For relying parties, it may be tempting to assume that the possession by a user of a medium level assurance certificate confers some additional authority on the holder (even though the relevant CP/CPS may make no such claim). It should be recalled that in the context of identity certificates, however, *assurance* hinges on two specific areas of activity:

— the care taken in verifying the identity of the user before certification;

— the care taken in managing the CA's internal operation, and its procedures for activities such as certificate rollover and revocation.

Nothing other than the certified identity of the holder is assured. Arguably, for many institutions, the RA mechanisms described here already satisfy the user authentication requirements of medium level assurance:

— Institutional databases are the source of authoritative information on staff and students, and provide secure and reliable storage for this data.

— Students supply personal information and proof of attainment before qualifying for entry to an institution; user credentials are normally supplied face-to-face to students in the process of induction.

— Staff undergo interview, examination of references, and inspection of professional qualifications before appointment. Institutional credentials are only assigned subsequent to the satisfactory completion of these checks.

In many institutions, both staff and students are issued with photo-ids on completion of their process of induction. It makes little sense to regard the registration process used to issue a photo-id as inferior to one that subsequently relies upon it.

Institutions, therefore, already take great pains to authenticate staff and students and these processes may be considered to satisfy this aspect of medium level assurance. A key point is that it is difficult to steal the identity of a specific user in the ac-PKI environment described here. The most effective lines of attack may be to find the careless user in the lab, or the unattended machine in an insecure office, but these human factor issues are a concern for almost any authentication scheme.

If an institutional RA and the CA are prepared to adopt the other mandatory aspects of medium level assurance (rekeying policy, revocation time, audit, private key storage) they could claim to satisfy all requirements of medium level assurance.

16.6 Technical issues

A number of technical issues exist that must be considered in the context of the properties and mechanisms required for the chosen PKI solution.

a) *Enrolment method*. The process by which a user obtains a certificate from the CA may involve either client-side or server-side generation of key pairs. In both cases, a variety of technical options exist for exchange of certification data between the user and the CA. In TIES, server-side key generation was implemented, as this appears the obvious way to provide a roaming server capability that enables the user to download the certificate and key at will. As hardware tokens become affordable, this is likely to affect the choice of enrolment method.

b) *Single key model.* The scope of the project is limited to authentication, and therefore requires only a single key pair for each user to provide digital signature capability (an additional key pair would be required to support encryption). Strictly speaking, there is no absolute requirement for the recovery of digital signature keys (unlike encryption keys), so where key recovery mechanisms have been discussed here, this is more a matter of user convenience than service necessity.

c) *Revocation method.* Any certificate, once issued, may subsequently have to be revoked (say, if the user leaves the institution). The revocation issue is considered significant in many descriptions of PKI deployment, and a variety of solutions have been considered, based on CRL and OCSP technology (see A.9). The first goal is to minimise the number of revocations issued:

   (i) for users who lose their keystore password, provide a recovery mechanism that does not involve revocation of the old certificate (either recover the original certificate and private key, or issue a new certificate and key without revoking the old one);

   (ii) select a validity period that coincides with the routine arrival and departure of users (i.e. the academic year), so that most certificates can be allowed to expire naturally.

   Secondly, the choice of CRL or OCSP (or both) must be made. On present evidence, demand for OCSP is weak and it is unlikely to be a stated requirement of relying parties in the foreseeable future.

   Thirdly, a variety of options for CRL publication exist: single or multiple distribution points; complete or delta CRLs, on- or off-line CRL fetching; distributed or centralised CRL dissemination. Under the most favourable circumstances, with a small number of CAs (preferably one), and a modest number of revocations, the more simple of these options (centrally managed distribution points for modestly partitioned, per-institutional CRLs) may be sufficient.

16.7 Preconceptions

A number of preconceptions were brought to the project, some of which have been confirmed, and others found false:

a) *PKI technology is mature.* Unfortunately, this appears to be false. There are very few PKIs deployed in European national academic networks. There are more examples of deployment in the commercial sector (particularly in the US), though interoperability problems between PKI implementations from different sources have long been encountered. While there are indications that this is improving (US Federal requirements have done much to iron out the flaws), caution is still required in the selection of a PKI vendor.

b) *Institutional CA implementation is straightforward using open source software.* Implementation of the basic technical components is relatively simple, as the project has demonstrated, but this forms a small part of the overall requirement. Anecdotal evidence from the development of the UK e-Science Grid CA indicates that the choice of open source may be a false economy, given the volume of effort required to make the software serviceable. This assertion, therefore, appears false.

c) *The main issue for the CA is getting the core technology right.* This is certainly important: with complex technology of this type there are numerous pitfalls that may only become apparent following deployment when problems arise in certificate validation and certification path processing. There are, however, numerous organisational, management, and human factors to be addressed (for example, the need for better private key management). Some vendors claim that the TCO of an in-house PKI is high even if their high-quality CA software is given cost-free. The main issue, therefore, is to get the whole solution right.

d) *CRLs solve the revocation problem*. In principle this problem was solved long ago. In practice, the industry has been slow to embrace the CRL as an effective mechanism for checking certificate status (in the first major revision of CRL format, some 10 years ago, the version number was not changed, since there were virtually no implementations of the original specification). There are still some signs of a lukewarm attitude (Microsoft certificate verification ignores CRLs by default). The uptake of the alternative OCSP solution has been slow (it's really only CRL by another interface). It appears that, with some reluctance, the industry has accepted that CRLs do solve the problem of revocation, as this project has also concluded.

e) *Browser management of certificates and private keys is straightforward.* Users can be given a simple set of instructions that, if followed correctly, will enable them to download and install a certificate on their browser. Beyond this, there are many limitations and pitfalls, and sites which have undertaken full PKI deployment tend to publish lengthy FAQs to assist their users in handling certificates on their browsers. User Support services must be prepared to offer the assistance and advice on certificate management that users will inevitably require.

17 Recommendations

The following recommendations are made for consideration by the JISC.

17.1 Adoption of technology

JISC must weigh the available evidence and make a decision on whether to move forward with an authentication solution for the H&FE community based on PKI. This is by no means a foregone conclusion, as alternative approaches are possible for each of the relevant application areas:

— *JISC Information Environment*. A number of solutions may be available, based on the use of existing password-based authentication schemes. These include Shibboleth, or name-password over SSL, possibly in parallel with a revised version of Athens.

— *The e-Science Grid*. Further development of the existing UK e-Science PKI could be left as a matter for the e-Science community.

— *Local institutional services*. Institutions are already considering the deployment of a variety of reduced-signon mechanisms to minimise the number of user logins. This could be left as a local matter for each institution. Equally, the ad hoc arrangements that institutions employ to issue digital certificates for server-side SSL authentication could also be left as a local issue.

The PKI solution, however, has two significant advantages:

— Web technology has become the predominant service-delivery method in H&FE, and PKI authentication is already in place in all the mainstream browsers and web servers in use in the sector. While the infrastructure would have to be delivered, there are no ongoing development costs.

— PKI is the only solution known to address all three application areas. The presumptive advantages of a unified PKI solution are cost, vendor-independent industry-standard technology, international application, compatibility with different authorisation schemes, and a strategic solution for authentication.

There are few reasons to expect that an ideal solution is in the wings, and the consequence of delay is the continuing diversification of authentication methods which would act as a barrier to the eventual deployment of a unified solution.

Recommendation 1: Adopt PKI as the preferred unified solution for authentication in the JISC IE, in the e-Science Grid, and for local institutional services.

17.2 CA architecture

The project implemented and tested each of the major PKI components using open source and Microsoft technology. In the case of the CA, both implementations were found to be practical at a technical level, but not adequate for the delivery of best organisational, management and operational practice.

Recommendation 2: Base the model for the ac-PKI on a vendor-supplied central Certification Authority, with a set of institutional Registration Authorities.

17.3 Vendor evaluation

The costs and capabilities of the main PKI vendor products should be evaluated to determine:

— the costs of the alternative approaches of an insourced solution based on licensed software, and an externally-managed outsourced solution;

— the quality of support for institutional RAs.

The corresponding costs of maintaining the present heterogeneous arrangements for authentication should also be considered.

This investigation may be managed by an ongoing project in the AAA Programme. The work itself may be undertaken by enlisting expertise in the community, or by recruitment of an external consultant, and should be completed promptly.

Recommendation 3: Undertake further evaluation of commercial PKI products to determine the relative costs of insourced and outsourced solutions, and the quality of support for institutional RAs.

17.4 Deployment plan

A staged deployment is necessary. Little more is likely to be discovered by further demonstrator-scale scoping studies, such as TIES itself. The hard issues are management buy-in, policy, and large-scale software engineering for robustness and high availability, none of which is well addressed by a small-scale research project, and none of which will be resolved other than by live deployment on a service-level scale.

The next step, therefore, is to undertake a full-scale deployment in a small number of institutions, reasonably representative of the various institution types in the sector. This would identify issues to be resolved before progressing to national deployment, and would develop roadmaps for similar institutions to follow. Participating institutions would undertake to issue certificates to all staff and students, and implement certificate-based access for some existing local password-based services.

This first deployment stage can offer institutions the advantages of expert support and possible funding to act as 'early adopters', but institutions are unlikely to participate unless there is a clear commitment to a full national deployment on a set timetable.

Recommendation 4: Plan a staged deployment of the ac-PKI, with a first stage to identify any issues specific to UK use and provide roadmaps for other institutions to follow.

17.5 Engagement

Successful deployment depends on proper engagement with all the parties concerned:

- The staged deployment should promote adoption by institutions, by producing cases studies which demonstrate the (modest) demands on institutions required for RA operation.
- The primary DSPs used by the participating institutions should be offered assistance in deploying certificate verification software to ensure that the bulk of JISC IE services accessed by the early adopters are equipped to handle certificate-based authentication.
- For JISC IE services no complex authorisation requirements have arisen, and the existing authorisation data used by Athens could be used by the ac-PKI. Discussions should be opened with EduServe to consider options for Athens interoperation/migration (for example, the Athens Authentication Point could be updated to accept certificates and enable certificated users to access all SSO-enabled DSPs).
- For Grid VOs, more complex authorisation requirements may be addressed by other means, such as the use of X.509 attribute certificates.

Recommendation 5: Engage with institutions, and with service providers in the ac-PKI, including JISC-funded centres, commercial publishers, EduServe, e-Science Grid, and the Common Information Environment.

ANNEX A – TECHNICAL REALISATION

A.1 Overview

Technical investigations were made in the following areas:

a) *Institutional RA*. Implementation of an institutional RA using data derived from institutional databases.

b) *Unix CA*. Implementation of a CA using Unix/Apache server technology. The two possible applications for this scenario are:

   (i) implementation of an institutional CA;

   (ii) implementation of a national JISC CA.

c) *Microsoft CA*. Implementation of an institutional CA using Microsoft Certificate Services technology.

d) *Unix certificate verifier*. Implementation of certificate verification in a standard DSP based on Unix/Apache. This is the common platform for most JISC IE services.

e) *Microsoft certificate verifier*. Implementation of certificate verification in an institution providing web-based services hosted on Microsoft IIS.

f) *DSP authorisation*. To demonstrate the integration of PKI-based authentication with a variety of authorisation technologies.

g) *Browser*. Testing certificate handling in the mainstream browsers for standard desktop equipment (Windows, Unix/Linux, Mac OS).

h) *Revocation*. Further consideration of revocation methods.

In retrospect, it is clear that a notable omission from this list is evaluation of commercial CA products. A study in this area would be a useful supplement to this report.

The findings in each of these areas of investigation are presented in A.2-A.9. A simple model of the interactions between RA, CA, and the user, is given in annex B.

A.2 Institutional RA

The Registration Authority needs to implement mechanisms that process data held by an institution about its users in order to facilitate the timely issue of certificates and their revocation. An institutional RA will carry out the following activities:

— Issue registration requests (register, revoke) to the CA

— Audit RA activity

It is important that the new processes required to support the creation and management of certificates are consistent with existing institutional processes. This will help reduce the barriers to take up and minimize the costs to the institution. The aim of the TIES RA has been to implement a system that can be incorporated into the existing arrangements for providing credentials to access local services and so can co-exist with systems supporting these local services.

To support the RA activities of registration and revocation, access to information about an institution's users is required. The principal members of an institution are its staff and students, and it is expected that well-developed processes exist to provide these users with credentials for local services. The information held about these two distinct categories of user will be 'owned' centrally within the organization although different groups may hold and manage the data for each.

Access to Institutional databases

TIES has anticipated there will either be onward flow of an institution's user data to other groups within the organisation wishing to use it or that there will be direct access to the centrally maintained database. Whichever method is used, a secure, authenticated, connection will be required for transfer or collection of data by the RA.

Data required by TIES

The data held on students and staff may, for organizational purposes, be slightly different. The following common data items were used in TIES:

- Unique identifier: for example staff personnel number or student enrolment number.
- Personal name: provides an additional user-friendly identifier (though not necessarily unique) for the user.
- Local username: used to access local services.
- Passphrase: used in conjunction with the local username.
- External username: a pseudonymous username intended for external use.
- Valid from: data for new students and staff may appear in the institution's databases before arrival and this may need to be reflected in the certificate validity.
- Valid until: the options regarding validity period are discussed in 11.3.2.

All of these items may be generated and stored in the institution's master databases. Alternatively, they may be generated and maintained by the RA. In time, it would be expected that all data would migrate to central databases.

A simple administrative web tool for managing the RA database was developed.

Timeliness of data

The timeliness of the user data may vary between institutions. It is anticipated that an RA will process data at least daily but may be required to do so more frequently. This frequency will need to match that used to maintain local services to ensure consistent provision across both local and external services.

Updates and revocation

Once populated, the TIES RA uses a file containing change information to maintain the currency of its database. These changes will reflect information about new and departing students and staff, or changes affecting the enrollment of existing members. If there are no existing institution processes that produce this change file, then the RA may generate one by comparing the latest presented data on staff and students against its own database. In the event of a user departing before the natural expiry of their certificate, the RA will issue a revocation request to the CA which will subsequently announce the revocation in the next CRL issued.

Other users

As has been noted, the processes used to support and provide services to staff and students are well developed, since these are required for the provision of local services. An institution may also wish to offer services to users who are neither students nor staff, but who may be categorised as visitors, such as visiting academics, honorary academics and students from other institutions where there are joint teaching arrangements. The organisational processes required to support visitors are likely to be informal and vary widely between institutions. As a consequence, this category of user has not been considered in the implementation of the RA developed in the project, although in some institutions the existing arrangements for visitors may allow them to be accommodated.

The RA disseminates institutional data as shown in figure A.1.

Audit

An audit trail is maintained by the RA. The actions that prompt an entry to be made in the audit file are:

- user registration request
- revocation request

Each entry in the audit file contains the information presented in the corresponding action request, plus the following:

— a timestamp for the activity recorded

— the reason for the entry

This records sufficient information to allow an institution to verify the status of any user certificate and the date of any change. It is envisaged that the RA audit records would be kept for a minimum of three years.

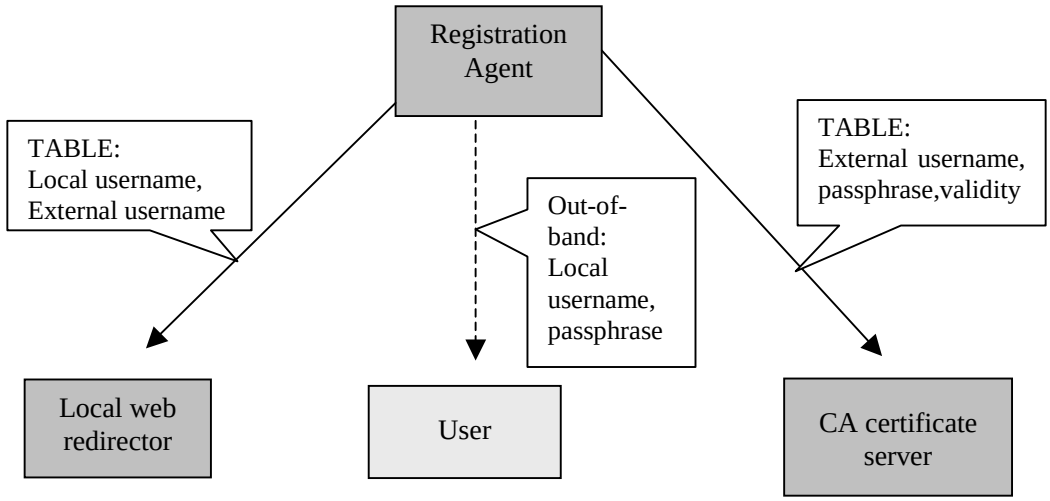

Figure A.1 — Overview of RA data dissemination

A.3 Unix CA

This activity comprised the following:

a) *Component identification*: Identify possible Open Source components for cryptographic and database functions.

b) *Systems design*: Certification processes.

c) *Systems integration*: Technical realisation.

d) *User registration*: A mechanism for bulk import of user registration and revocation data from the RA.

e) *Certificate granting*: A mechanism to respond to a certification request issued by a user.

f) *Administration*: Certificate revocation and status review.

i) *Audit*: To record all registration/revocation/certificate granting events.

A.3.1 Component identification

The strategy was to identify possible open source solutions and then spend some time to make an assessment of each product. For those considered likely candidates, a build and install was attempted.

The following open source solutions were considered:

a) *OpenCA*

The former OpenCA Project now has two threads:

(i) OpenCA PKI development at `sourceforge.net/projects/openca`. SourceForge is owned by Open Source Development Network, Inc., which is a wholly owned

subsidiary of VA Software Corporation (http://www.vasoftware.com). The latest version is 0.9 (Beta) released 12/9/02.

(ii) OpenCA LABS at www.openca.org. The OpenCA PKI Development Project is a collaborative effort to develop a robust, full-featured and Open Source out-of-the-box Certification Authority implementing the most-used protocols with full-strength cryptography world-wide. The latest version is 0.9.1 (presumably Beta) released 7/1/03.

Both OpenCA PKI and OpenCA LABS are based on OpenSSL and OpenLDAP with Apache and mod_ssl. (mod_ssl is required for this implementation because it uses client generated key pairs). Neither has an OCSP responder but have placeholders. The OCSP responder project for OpenCA PKI was registered in November 2001 but has yet to release any files.

The build and install were relatively straightforward although the documentation was patchy. Use of an LDAP directory is optional – the default Berkeley database (DB) was used instead for simplicity. The initialisation of the CA and RA, and generation of the CA key and CA self-signed certificate was successfully performed using the web interface. A user certificate request was made, approved by the RA, imported into the CA and issued to the user. Because the TIES project used a local web-director to keep usernames pseudonymous, rather than try to change the somewhat complex structure of the OpenCA front-end a much simpler user interface to OpenSSL would be undertaken.

b) *pyCA* at www.pyca.de

This is a Python based implementation of a CA. There is no OCSP support. This project seems to have stalled - there is no on-going development at this time just bug fixes. The last release was 0.6.6 (date unknown but probably in year 2000 – 0.6.5 was released in July 2000).

c) *MyCA* at sourceforge.net/projects/myca

This claims to be a "Multi-CA solution with support for complex certificate chains." This is basically a shell script interface to OpenSSL and has no web interface or OCSP support. The latest version is 0.4 (Beta) released June 2003.

d) *getronics* now at www.getronicsgov.com/knowledge/cml_home.htm

This is a freely available Certificate Management Library (CML) but the problems of building the library from its source components proved too time-consuming to be worth pursuing.

Getronics (Getronics Government Solutions) was acquired by DigitalNet in the third quarter of 2002 and the original URL (www.getronicsgov.com/hot/cml_home.htm) is now a dead link.

The current, re-packaged version now builds successfully (after fixing a glitch in a generated `makefile`) but as this is just a library; more front-end code would need to be developed and the API properly understood before a reasonable test could be made. OCSP support is on the list of "things to do". The latest version is 2.2, released in February 2003.

The licence for this library is provided by 'The United States Government/Department of Defense/National Security Agency/Office of Network Security (collectively "the U.S. Government")', which might raise some eyebrows.

e) *cryptlib*

This is a Security Toolkit that was found easy enough to build but not to use. For example, the self-test program failed during testing certificate creation/export; remaking cryptlib with debugging turned on results in an internally trapped error. There is, however, a commercial version, which may be easier to use. Although cryptlib makes use of some OpenSSL code, it is not simply an interface to the standard OpenSSL library. Instead it contains embedded copies of some OpenSSL source code (see the acknowledgement in the user manual, p.260). How it intends to synchronise with ongoing OpenSSL bug fixes is not clear. The latest version is 3.1 beta 2, released in September 2002.

f) *OpenSSL*

OpenSSL is used in some way by all of the open source solutions that we considered. It is a stable, well-proven product but the documentation is somewhat sparse and mostly consists of the manual pages and

contributed Perl documentation. That said, there is a great deal of OpenSSL experience out there and a web search has, so far, always provided a pointer or answer to every question that we have had.

None of the open source products in themselves provided a complete, satisfactory solution for our requirements — only OpenCA and pyCA have a user interface and the need for pseudonymous naming would in any case have required changes to this. It was concluded that the best way forward was to write a simple, web-based front end to OpenSSL.

For revocation checking, OpenSSL includes a prototype OCSP responder that is sufficiently robust for testing purposes. This was adapted to interrogate the Postgres database for revocation information (see A.3.6).

The use of LDAP was considered but thought to be over-complex for the immediate requirements of the TIES project. Instead Postgres, a relational database management system (RDMS), was used as the certificate store. The same database was also used to hold audit data and other material. OpenLDAP uses the Berkeley DB concurrent/transactional database which is considered to give greater performance and scalabilty than an RDBM. It would be possible to use Postgres as a backend to LDAP.

An important consideration with any open source product is how rigorously it has been tested. Does it come with a test suite? Can it be trusted? In general, trust in an open source product is slow to develop, and emerges only after extensive use and peer review. OpenSSL is widely used but it is essential to ensure the integrity of the version that is obtained by checking the supplied MD5 checksum or PGP signature.

A.3.2 System design

The Registration Agent maintains the following for each user:

- a local-username,
- a passphrase,
- an external-username,
- a validity period.

It hands off this information as follows:

- The user is given, out-of-band, the local-username and passphrase.
- The CA certificate Server is sent, for each user, the external-username, passphrase and validity period.
- A Local Web Redirector is sent, for each user, the local-username and corresponding external-username.

When the user wants to request a certificate, the steps are as follows:

- The user follows a link that points to the Local Web Redirector.
- The Local Web Redirector asks for the local-username and sends a redirection instruction containing the corresponding external-username and the institution name.
- The redirection takes the user to the CA Certificate Server. If the external-username is recognised (and is still valid) a passphrase challenge is issued. If the user supplies the correct passphrase a certificate and key are issued with a subject name formed by concatenating the external-username and institution. (Alternatively, the RA may send the CA the subject name as an additional field in the registration.)

This approach has the useful property that the user is only required to know the local-username and passphrase. Since the local-username may, in some cases, supply identifying information, only the external-username is sent off campus. This comes at the price of having to maintain the Local Web Redirector. See figure A.2.

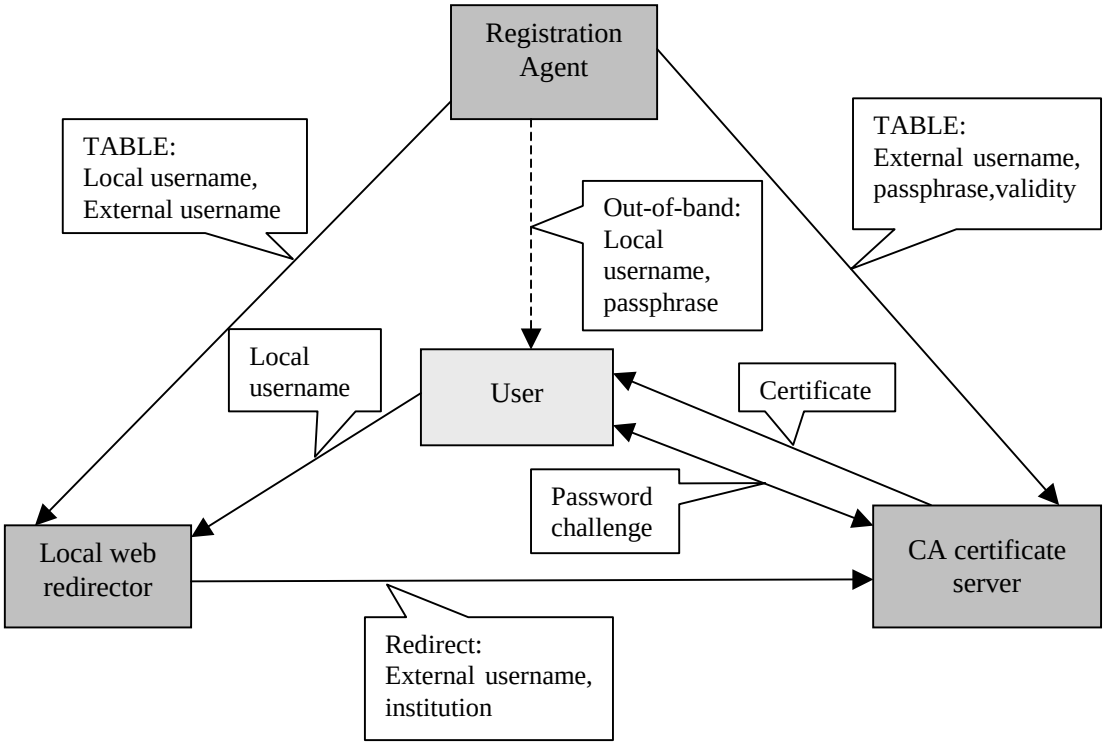

Figure A.2 — Simple TIES model

A.3.3 System integration

The end-user-facing part of the CA was based on three main components:

- *OpenSSL* for the basic CA functionality
- *Postgres* for database functionality (certificate store, audit log, etc.)
- *Apache* web server to support certificate requests from browsers

OpenSSL includes a complete but basic CA implementation. It is command-line driven. The example below creates a new, empty CA, with a new private key, then generates a certificate request and signs it. The output is a base64-encoded (PEM format) certificate file. The certificate request stage takes the distinguished name of the certificate subject as input:

```
CA.pl –newca
CA.pl –newreq
CA.pl –signreq
```

`CA.pl` is a simple script supplied with OpenSSL that hides some of the details of the underlying `openssl ca` commands from a beginner. It uses hardwired filenames like `newcert.pem`, which is why no filenames appear above. The underlying code is extremely flexible though. It puts practically all of the properties of the generated certificates under the user's control, via a comprehensive configuration file. In some contexts this may be a significant advantage when compared to closed commercial software.

The OpenSSL CA tools described above are sufficient on their own to implement a small CA where the administrator can manually generate each certificate. This approach would be adequate for a CA issuing small numbers of certificates for local SSL web server machines, for example. Extending this to allow larger numbers of untrained users to request certificates remotely using web browsers is fairly straightforward. Because the CA tools are command-line based, they can easily be called from perl CGI scripts invoked by Apache-hosted HTML pages. This became the basis of our CA implementation.

Other than the lack of a web front end, the main limitation of the CA implementation supplied with OpenSSL is its certificate store. This is implemented as a number of directories and flat files holding the CA certificate, the CA private key, copies of the generated certificates (named after their serial numbers) and so on. The "database" includes two files:

— `serial`, the serial number that will be allocated to the next certificate issued, as a text string

— `index.txt`, a flat text file holding records of the serial number, distinguished name and revocation status of every certificate issued, one per line.

These two files are read into memory at the start of every CA command, modified and written back. Obviously, this will not scale well to large numbers of certificates. However, OpenSSL makes no such claim and clearly states the limitations in this area. A full-scale CA based on OpenSSL must therefore implement its own database back end. Thankfully though, the text database is well isolated in a separate module, with explicit API calls used to read and update database records (see `crypto/txt_db/txt_db.h` in the OpenSSL source distribution). It was therefore relatively simple for us to build a variant version of OpenSSL that intercepted these calls and replaced the `txt_db` module with one that interfaced to a Postgres relational database.

A.3.4 Import of user information from the RA

The previous clause describes the CA components that implement actual certificate creation (OpenSSL), the web interface (Apache, plus locally developed scripts and HTML pages), and the certificate store (Postgres). The remaining components handle secure exchange of data with the RA.

A `cron` script does nightly downloads of current registration data from the RA, using standard Unix tools:

— *Secure copy (`scp`)* for bulk file transfers from the RA system.

— *Secure shell (`ssh`)* to provide encrypted data transmission, RSA authentication of the server machine to the client, and either key or passphrase based authentication of the client to the server.

Cooperation from the University of Edinburgh's Computing Services and Management Information Services departments allowed copies of the University's "live" registration database to be obtained. The size of this file motivated the design of additional code to send only data changed since the previous download, rather than copying the entire dataset every time. Even so, a cut-down database containing only the project personnel was used for much of the testing.

A.3.5 Requesting a certificate

After supplying the local-username and passphrase to login to the CA web server via the local redirector, as described in A.3.2, a user is presented with a number of options in the form of web links, as shown in figure A.3.

a) *Request a new certificate.* If the user has previously been granted a certificate, this is shown as *retrieve an existing certificate.* In either case, the user is prompted to supply another passphrase. This need not be the passphrase used to log into the CA; it is a separate envelope-passphrase used to encrypt the certificate and its corresponding private key in a PKCS#12-format file (Netscape `.p12`, Windows `.pkx`) for secure transmission to the browser. The possibility of user confusion here is acknowledged. After entering the passphrase, the CA server sends the PKCS#12 file to the browser. The user sees the browser's standard dialogue asking if they want to open the file or save to disk. In either Netscape or Internet Explorer, opening the file starts the process of importing the certificate into the personal certificate store, during which the user must supply the envelope-passphrase again.

   NOTE: In Internet Explorer the Certificate Manager Import Wizard is invoked but fails to import the data properly. The PKCS#12 file must be saved to disk before opening it manually. An actual deployment would have to either resolve this issue or undertake a significant amount of user education.

b) *Retrieve CA certificates.* Links are provided to download the certificates of both the institutional CA and the simulated JISC national CA. The user needs to download both of these before using their personal certificate. The same certificate import process is invoked as in (a), but no passphrase is required, since no private keys are being sent in this case. The CA web server simply returns the certificate data to the browser with the HTTP header `Content-Type: application/x-x509-ca-cert`. This automatically invokes the browser's certificate-importing process.

c) *Test installed certificate.* The version of Apache hosting the CA web server was built with SSL support (using mod_ssl) and could therefore be used to provide a check of certificate validity as described in A.5.1 and A.5.2. Example output is shown in figure A.6.

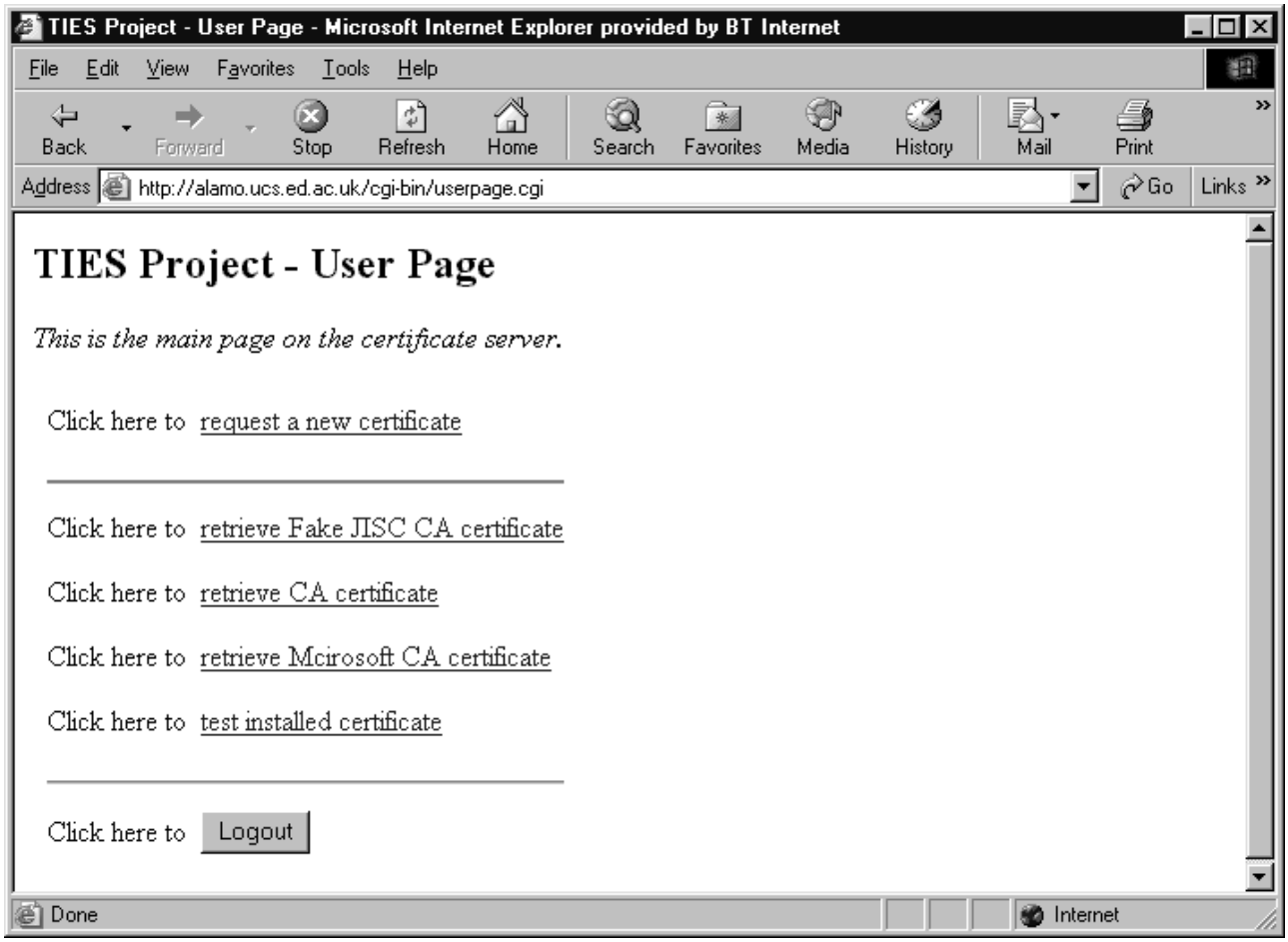
TIES Project - User Page - Microsoft Internet Explorer provided by BT Internet
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print
Address http://alamo.ucs.ed.ac.uk/cgi-bin/userpage.cgi
Go Links
TIES Project - User Page
This is the main page on the certificate server.
Click here to request a new certificate
Click here to retrieve Fake JISC CA certificate
Click here to retrieve CA certificate
Click here to retrieve Mcirosoft CA certificate
Click here to test installed certificate
Click here to Logout
Done
Internet

Figure A.3 — Unix CA web interface

A.3.6 Administration

In addition to the end-user interface described up to now, the CA web server also provides a separate, passphrase-protected administration page with the following options:

— *Revoke a certificate*

— *Display audit trail*

— *View various internal database tables*

The underlying technologies used to support revocation (CRLs and OCSP) are discussed in 11.2. When a certificate is revoked, built-in OpenSSL commands are used to update the certificate store and generate a new CRL from it. In practice, the CRL file should only be regenerated at regular intervals, say every 24 hours.

To allow relying parties to check whether a certificate has been revoked, the certificates generated by our CA contain:

— A URL that can be used to download the latest CRL.

— The URL of an OCSP responder process permanently running on the CA machine.

The OCSP responder process runs the appropriate `openssl ocsp` command and uses the same version of OpenSSL, modified to use Postgres for the certificate store, as the other CA components. It therefore has access to the live certificate store data and is always able to supply up-to-date status information.

A.3.7 Audit

The following extract shows the kind of records maintained by the CA for audit purposes:

```
2003-03-26 16:01:42+00 issue   ext-johnm@ties.org (jacinto.rcs.ed.ac.uk) #19 2004-03-25 16:01:42+00
2003-03-26 16:02:08+00 revoke  ext-johnm@ties.org #19 unspecified
2003-03-31 11:30:53+01 issue   ext-ian@ties.org (shortbread.rcs.ed.ac.uk) #20 2004-03-30 11:30:53+01
2003-04-03 17:24:37+01 issue   ext-sandy@ties.org (ua-3.rcs.ed.ac.uk) #21 2004-04-02 17:24:37+01
2003-06-04 15:53:50+01 issue   ext-kiz@ties.org (shortbread.rcs.ed.ac.uk) #22 2004-06-03 15:53:50+01
2003-08-20 15:12:34+01 issue   ext-johnm@ties.org (jacinto.rcs.ed.ac.uk) #23 2004-08-19 15:12:34+01
2003-08-29 10:03:38+01 issue   ext-alan@ties.org (braemar.rcs.ed.ac.uk) #24 2004-08-28 10:03:37+01
```

For each certificate issued, the serial number, the external-username, the DNS name of the client and the expiry date are shown. For revocations, any of the standard reasons may be given (“unspecified” above). There are relatively few entries because the system records the generation (issue) of new certificates rather than the downloading of certificates, which may happen multiple times to different machines.

A.4 Microsoft CA

One of our objectives was to investigate the extent to which the out-the-box Microsoft certificate mechanism could generate and administer certificates with the necessary properties to access JISC IE services.

At the outset of the project, Windows 2000 Server was the most widely deployed Microsoft server operating system, so we adopted it as our baseline. Since then, Microsoft has released Windows Server 2003 (formerly .NET Server), which may be expected gradually to replace the previous edition. Looking at the on-line documentation for the new software indicates that changes in the certificate services area are incremental and evolutionary; we do not therefore expect our conclusions to be affected.

A.4.1 Installation

Microsoft Certificate Services, which includes the functionality of a certification authority, comes bundled with Windows Server 2000 at no extra cost. It is installed as a separate optional component, either during the main server installation or from Add/Remove Programs on the Control Panel afterwards. To enable certificate requests from remote browsers, administrators must install Microsoft’s IIS web server before Certificate Services (see A.4.2 below).

Two main choices must be made when installing Certificate Services:

— An installation can be “Enterprise” (requires Active Directory) or “stand-alone” (does not require Active Directory, but will use it if present).

— The CA may be a “root” (generates its own self-signed CA certificate) or “subordinate” (obtains its CA certificate from a higher-level CA).

All four combinations are supported. We wanted our Microsoft CA’s certificate to be signed by a simulated national JISC CA, so we chose a stand-alone, subordinate installation. For the other (“advanced”) options, we used the default Cryptographic Service Provider (CSP) plug-in software, as well as the default hash algorithm, key lengths, etc. Other information that must be provided during installation includes the X.509 Distinguished Name for the CA and the name of a local directory to hold the generated certificates. We did not create the optional issuer policy statement.

When installing a subordinate CA, a certificate signing request is automatically generated and saved to a file in DER format. We were able to take this file to the Linux system running our simulated national CA, generate a signed certificate based on this request using the OpenSSL toolset, and export the PEM-format output file back to the Windows system. The installation was then restarted and the imported file successfully used as the new CA certificate. It was also necessary to import the simulated JISC CA root certificate into the Windows certificate store as a trusted CA, using Internet Explorer’s usual Tools, Internet Options, Content, Certificates, Trusted Root Certification Authorities interface.

A.4.2 Web-based certificate enrolment

Microsoft refers to the process of end users requesting and being granted certificates via web-based forms as "enrolment". The required forms and scripts are automatically installed along with Certificate Services if Microsoft's IIS web server software is present. The forms are installed at http://*server*/certsrv, where *server* is the DNS name of the Windows server machine. Figure A.4 shows this page on our test system. As you can see, the form allows users to perform various certificate-related tasks, the main ones being to request a certificate and then check to see if it has been granted. If a certificate has been granted, it can be downloaded to the client's browser.

Microsoft Certificate Services - Netscape
File Edit View Go Bookmarks Tools Window Help
http://jilin.ucs.ed.ac.uk/certsrv/
Search
Mail Home Radio Netscape Search Bookmarks
My Sidebar
Search
Search Results
Bookmarks
Buddy List
Shopping
Netscape Channels
Movies & Music
Maps & Directions
Microsoft Certificate Services -- TIES Microsoft CA
Home
Welcome
You use this web site to request a certificate for your web browser, e-mail client, or other secure program. Once you acquire a certificate, you will be able to securely identify yourself to other people over the web, sign your e-mail messages, encrypt your e-mail messages, and more depending upon the type of certificate you request.
Select a task:
Retrieve the CA certificate or certificate revocation list
Request a certificate
Check on a pending certificate
Next >
Document: Done (13.18 secs)

Figure A.4 – User requests a certificate from Microsoft Certificate Services

NOTE: The attentive reader may have spotted the use of Netscape 7.0 rather than Internet Explorer in figure A.4. As a side-effect of a security fix, Microsoft have managed to get themselves into a position where, depending on the relative level of patches applied to server and client, their certificate request web form works with Netscape but not with Internet Explorer. The symptom is the appearance of a message, "Downloading ActiveX Control," in the Internet Explorer window, which then hangs. The causes, and the work-rounds (applying consistent patches), are described in detail in:

http://support.microsoft.com/default.aspx?scid=kb;en-us;330389

One can hope that by the time certificates come to be widely deployed, this has settled down. In any case, we were not greatly worried, since a real site would be used to this kind of versioning issue, and for locked-down, standardised client desktops for enterprise deployment, could ensure the situation did not arise. For ourselves, we simply used Netscape.

A.4.3 Granting certificates

There are two ways of granting a certificate requested via web enrolment:

a) An administrator can review pending requests and manually grant or refuse them using a graphical tool.

b) The system can be configured to grant requests automatically.

Enterprise installations always process requests automatically. The expected mode of operation is to use standard IIS security mechanisms to protect the web enrolment pages, so that they are only accessible to users who have logged in to the local Windows domain. Since only logged-in users have access to the enrolment pages, IIS can automatically derive the distinguished name for the certificate from the Active Directory entry for the Windows username (in stand-alone mode, the user must manually enter their own distinguished name into the web form). The administrator can mark hierarchies of users within Active Directory as entitled to request certificates (or not).

The net result is that in this mode, practically no additional administration effort should be required to issue certificates to large numbers of end users, provided they already have login credentials on a system in the local Windows domain.

In a stand-alone installation, by contrast, the administrator must periodically check for certificate requests from end users by running the Certificate Services administration tool. Pending requests appear in a “Requested Certificates” folder. The administrator must manually inspect the outstanding requests, which include the requested certificate details such as distinguished name, and for each one either grant the certificate or reject the request. The administrator is assumed to have sufficient time and out-of-band knowledge to be able to match certificate requests to real users and reject requests from, for example, “Mickey Mouse, Disneyland”. Once granted, a certificate is available for downloading the next time the user polls the status of their request. Note that because of the manual intervention required, there is likely to be a significant delay between a user requesting a certificate and the request being granted. Clearly, the stand-alone mode is intended for small departments and would scale poorly to use in an institutional CA, for which an Enterprise installation would be more suitable.

The graphical certificate administration tool (figure A.5) is basic, but still a welcome change from the OpenSSL command-line equivalents. It supports certificate granting, revocation, backup and restoration. The latter three functions are equally applicable to Enterprise installations.

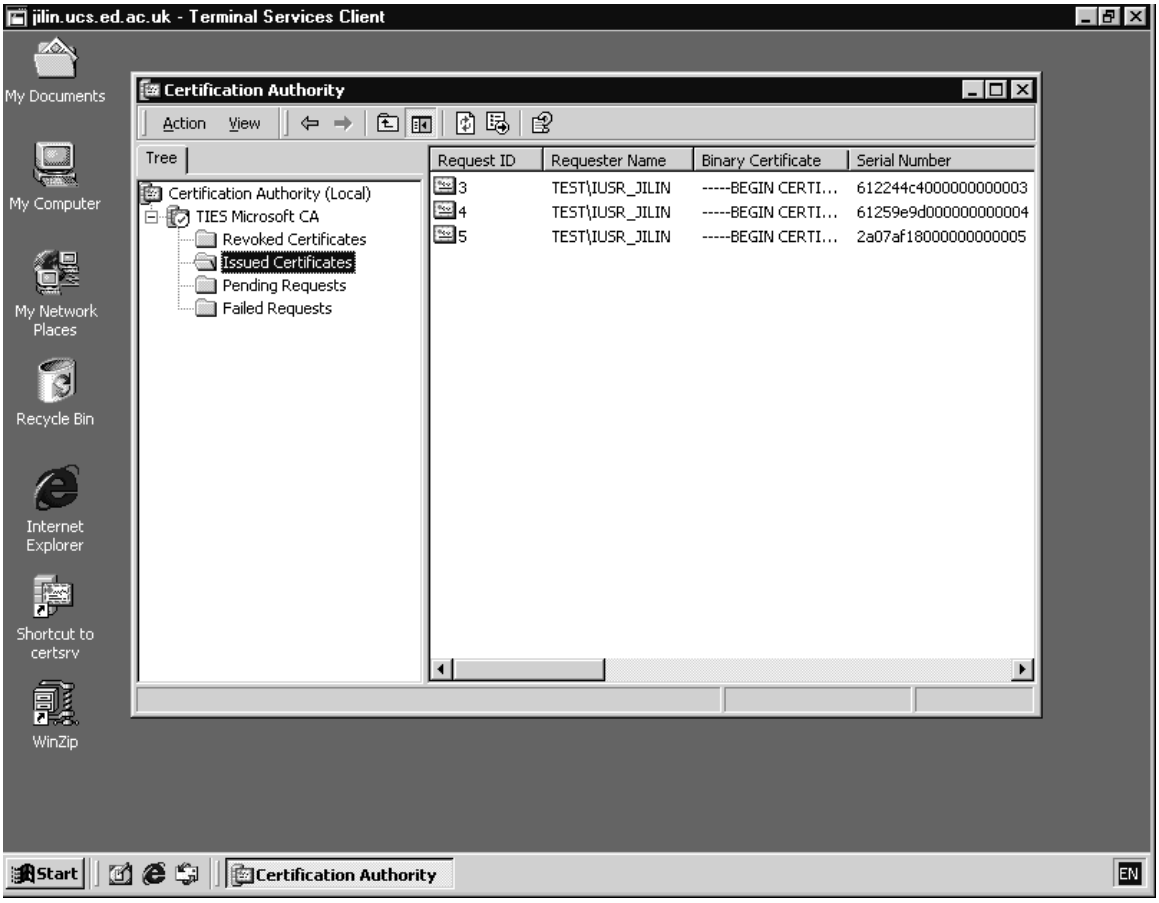

Figure A.5 — Microsoft Certificate Services administration GUI

A.4.4 Certificate compatibility

Our main interest was in the question of whether or not certificates issued by the Microsoft CA were compatible with Apache web servers running mod_ssl client authentication, for possible use in the JISC information environment. We were pleased to discover that the certificates were indeed compatible. Figure A.6 shows a

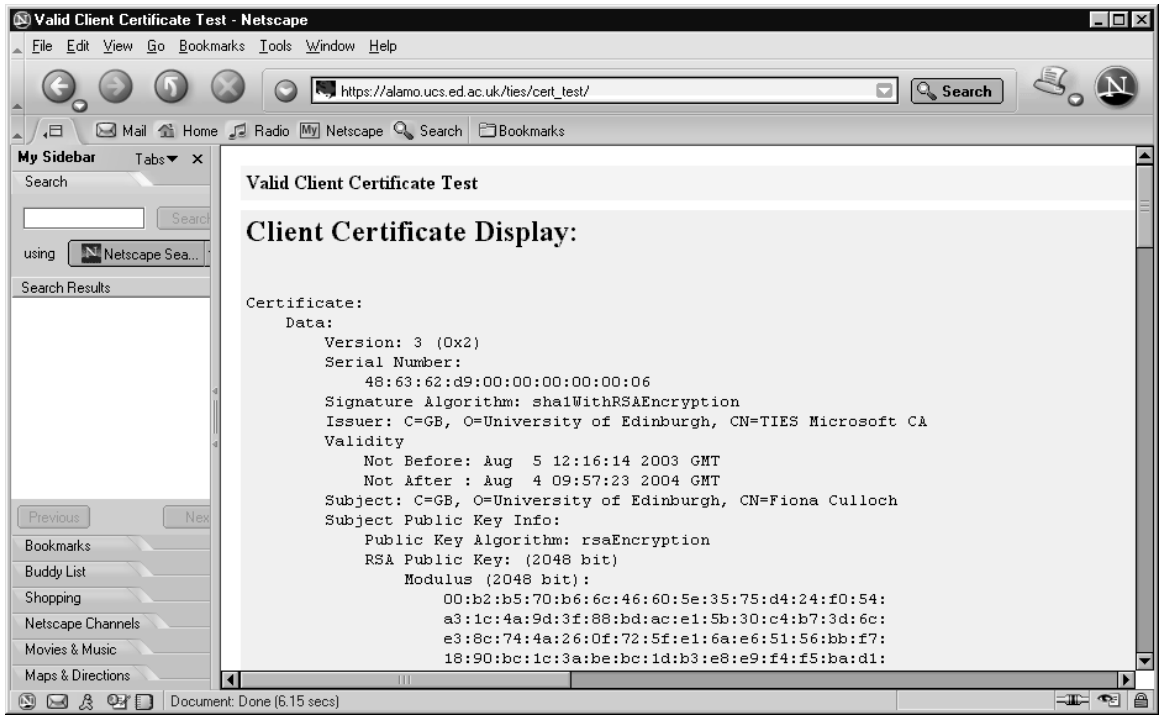

Figure A.6 — Linux test page accessed with certificate issued by Microsoft CA

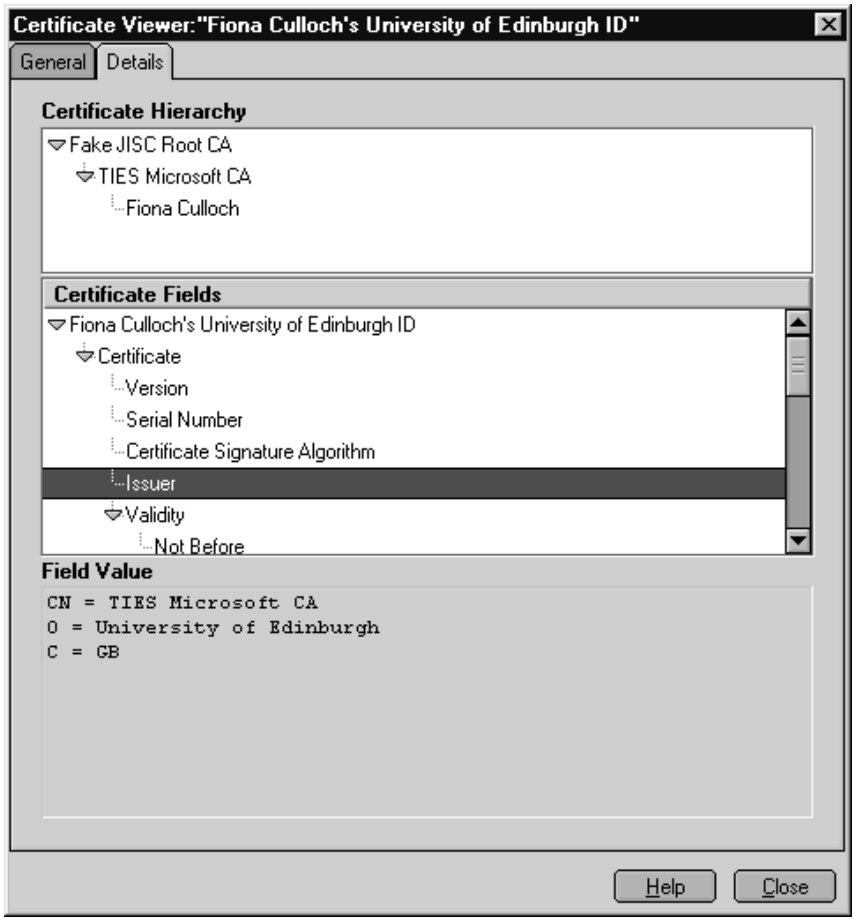

Figure A.7 — End-user certificate showing CA hierarchy

Microsoft certificate being used to access the certificate test page of our Linux-based institutional CA. The certificates were also usable to access our test JISC IE service.

Note the long (80-bit) serial number, which is clearly some kind of globally unique identifier, with the least significant bits being an actual serial number. This differs from OpenSSL certificates, which use simple serial numbers but the difference does not seem to cause problems: the certificate decode shown is in OpenSSL format.

Note also that the Microsoft certificates are properly subordinate to the national JISC CA. This is implicit in the fact that the certificate can be used to access the test site, but can be seen explicitly in the certificate hierarchy shown in figure A.7.

A.4.5 Certificate revocation

Clause 11.2 describes the two methods we investigated for data service providers to check whether or not a presented certificate has been revoked since it was issued: a Certificate Revocation List (CRL) published by the CA, or the Online Certificate Status Protocol (OCSP). The OCSP method is newer, sufficiently new that Windows 2000 Server does not contain built-in support for it (neither does 2003, though the documentation contains signs of movement in that direction). Communication with a contact at Microsoft confirmed this but revealed the existence of a number of commercial third-party OCSP offerings:

- Alacris, http://www.alacris.com
- Ascertia, http://www.ascertia.com
- KyberPass, http://www.kyberpass.com
- Seguridata, http://www.seguridata.com
- Valicert, http://www.valicert.com

A free client was available from one company (Alacris) but no free server (responder), which is the component that would be required by an institution wanting its own CA.

On the other hand, the CRL method is well supported. Certificate Services automatically sets itself up to publish a CRL at configurable intervals. Likewise, the certificates it issues contain extensions indicating where to find the latest CRL:

```
X509v3 CRL Distribution Points:

URI:ldap:///CN=TIES%20Microsoft%20CA,CN=jilin,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=
Configuration,DC=test,DC=enterprise,DC=ucs,DC=ed,DC=ac,DC=uk?certificateRevocationList?base?obj
ectclass=cRLDistributionPoint

URI:http://jilin.test.enterprise.ucs.ed.ac.uk/CertEnroll/TIES%20Microsoft%20CA.crl
```

As can be seen, the CRL is accessible via either LDAP or HTTP.

A.4.6 Microsoft CA conclusions

Other work in this project has led to the conclusion that most institutions would be better off acting as registration agents for a centralised regional or national CA than each attempting to implement their own local CA. Nevertheless, there are bound to be some exceptions, and if these include predominantly Microsoft-based institutions then the work reported above gives some level of assurance that they would be able to use Microsoft Certificate Services unmodified with the JISC IE, provided that OCSP revocation checking is not specified as a requirement. See A.9.4 for more about the interaction of a Microsoft CA with various possible revocation schemes.

The other main question mark outstanding is that one might reasonably expect most institutions to want to use the "Enterprise" version of Certificate Services, while for simplicity we have evaluated the "stand-alone" version. We are relying on the assertion in Microsoft's documentation that the same feature (of saving subordinate CA certificate requests to files) is available for an Enterprise as for a stand-alone CA, to allow subordination to a non-Microsoft JISC national CA. Note that, because the distinguished names in Enterprise certificates are derived from Active Directory entries, they will normally contain users' real names, and therefore will not fully conform to the proposed TIES certificate profile (11.3), which mandates pseudonyms for privacy reasons.

A.5 Unix DSP certificate verifier

This clause covers the main areas of implementation that must be undertaken by a Unix/Apache-based data service provider (DSP) to integrate the use of digital certificates: certificate verification and the subsequent use of certificate information for authorisation.

A "real world" implementation using an existing EDINA service, called "Update", was done so that the software effort required could be realistically gauged rather than theoretically estimated. This actual implementation, using Apache under Sun Solaris, is outlined in this clause. Since there are no current EDINA services running under Microsoft IIS, a more artificial exercise was conducted to test certificate verification in the Microsoft context. That is discussed in clause A.6.

The TIES project's overall approach has been as far as possible to seek existing, proven software components. One such component is the well-known secure socket layer (SSL) protocol, which is normally used when a client (web browser) wants to assure itself that it is talking securely to a known, authenticated web server. This is the "https" protocol used for on-line purchases, banking and so on. A less well known aspect of the protocol is its support for *client authentication:* a server can require that clients connecting to it must present a valid X.509 digital certificate, signed by a trusted certification authority (CA). A client without a certificate, or whose certificate is invalid, or from an authority not trusted by the server, will be rejected. Our initial assumption was therefore that a good SSL web server implementation would automatically provide us with the ability to verify the integrity of certificates presented by users of JISC information environment services, and that our task would be to find out whether there was a commonly available SSL implementation that was up to the job.

A.5.1 Chosen implementation

One of the project's assumptions was that most web services in the JISC IE would be hosted by the Apache web server, http://httpd.apache.org. However, there are two branches of Apache development in common use: 1.x and 2.x. Apache 1.x does not provide built-in support for SSL but is still heavily used. In particular, it is used for all current EDINA services, making it the obvious choice for this project. Investigation indicated that the SSL add-on for Apache 1.x most commonly mentioned was:

— mod_ssl, http://www.modssl.org, which relies on

— OpenSSL, an underlying SSL library, http://www.openssl.org

This combination of components was therefore adopted. The mod_ssl `INSTALL` file provides a reasonable overview of how to download and install all three components (Apache 1.x, mod_ssl and OpenSSL) so that they work with each other.

A.5.2 Configuration

After installing mod_ssl, the following SSL-related statements were added to Apache's configuration file, `httpd.conf`, as described in the mod_ssl manual:

a) `SSLVerifyClient require`

This turns on client authentication. Clients must then present a valid certificate before they can access web content.

b) `SSLVerifyDepth 2`

In practice, a client browser presents not just its own certificate, but also the certificate of the CA who signed its certificate, and possibly of a higher authority who signed *that* certificate, and so on. The `SSLVerifyDepth` parameter controls how far up the *certificate chain* the server will search for a certificate on its "trusted" list (see below) before giving up and declaring the client's certificate invalid. A depth of 0 allows only self-signed certificates, 1 allows either self-signed certificates or certificates signed by an authority directly known to the server, and so on. Depth 2 matches our model of a national JISC CA that signs the certificates of participating institutions or regional certification authorities. For example, a personal certificate might be signed by Edinburgh University, whose certificate is in turn signed by JISC. A greater maximum depth would be required if institutions were to create further subsidiary certification authorities, for example at departmental level.

c) `SSLCACertificateFile ~/apache/conf/ssl.crt/ca-bundle.crt`

This file is just the concatenation of the individual certificate files of all the certification authorities to be trusted by this server. A client is accepted if its certificate is signed by an authority in this list, which as shipped with mod_ssl contains all the common commercial certification authorities (VeriSign, Thawte and the like). For the purposes of the JISC information environment though, we only want to accept certificates issued either directly by the national JISC certification authority or indirectly by a subsidiary institution. A DSP should therefore edit this file, replacing its contents with the certificate of the national JISC CA. Users with JISC-issued certificates, or with certificates from a subsidiary institutional CA signed by the JISC CA, will then be accepted; users with (valid) certificates from other certification authorities will be rejected. Note that the DSP does not need to know the certificates of hundreds of individual institutions.

NOTE: DSPs which need to accept non-JISC certificates for other purposes can have different trusted-CA lists for different Apache virtual servers. Alternatively, they can add the JISC CA to the default list of commercial authorities, but they will then need another mechanism to ensure that only clients with certificates signed by the JISC CA are permitted to access JISC information environment services. However, this project's background assumption is that the problem is getting sites to use certificates at all, not that existing uses of certificates will conflict with the JISC information environment.

d) `SSLOptions +StdEnvVars +ExportCertData`

The `ExportCertData` option, which by default is disabled, causes mod_ssl to export client certificates in PEM format to the `SSL_CLIENT_CERT` and `SSL_CLIENT_CERT_CHAIN_`*n* environmental variables for use by server-side scripts. Our certificate-enabled service requires this (for OCSP handling only).

A.5.3 Integration with data services

A certificate-based login procedure was added to an existing EDINA service, "UPDATE", which contains bibliographic references on farming, the countryside and the environment for UK HE & FE users. A full description of the process would necessarily include much material that is specific to the particular methods EDINA uses to create user sessions and handle logins. We will thus only describe necessary changes in general terms as a demonstration that certificates can be added relatively easily to a web site not designed with that in mind.

A.5.3.1 User interface

UPDATE, like most EDINA services, has a main login page that shows current service status and provides two buttons for logging in: one for academic users, which invokes Athens single-sign-on, and one for non-academic (commercial) users, based on a simple encrypted password file held locally. The only interface change required was the addition of a certificate-login button linked to an https URL (see A.5.3.8). The result is shown in figure A.8.

A.5.3.2 Concurrent session management

The result of logging in is a new database session. The EDINA servers keep track of multiple users' state (searches in progress, results retrieved, filters set) by means of a session identifier embedded in each URL presented by the browser, e.g.:

`http://`*server*`.edina.ac.uk/WebZ/html/search.html?sessionid=01-61535-1419431530`

All action links embedded in each displayed page contain the session identifier, allowing the server to associate new requests with the user's previous context. Other implementations are possible (e.g., cookie based) and DSPs are likely to differ in this area.

A.5.3.3 The standard login script

The standard login procedure for EDINA services, using Athens authorisation, fulfils a number of functions. It:

— calls Athens to check that the user is authorised for this particular service

— creates a new database session for the user

— captures the HTML output from the database (normally the initial search page) and echoes it to the user, with relative links massaged to function in the new context

— writes a time-stamped record of the user's access to the service into both a local log file and a central Athens log.

These functions are obviously common to most services, and are factored out into a shared package, `Edina::Autho::LoginScript`. It was desirable that Athens authentication could be replaced by certificate acceptance with minimal changes to existing code. In fact it proved possible by overriding only one function (`authorise`) as described in A.5.3.6.

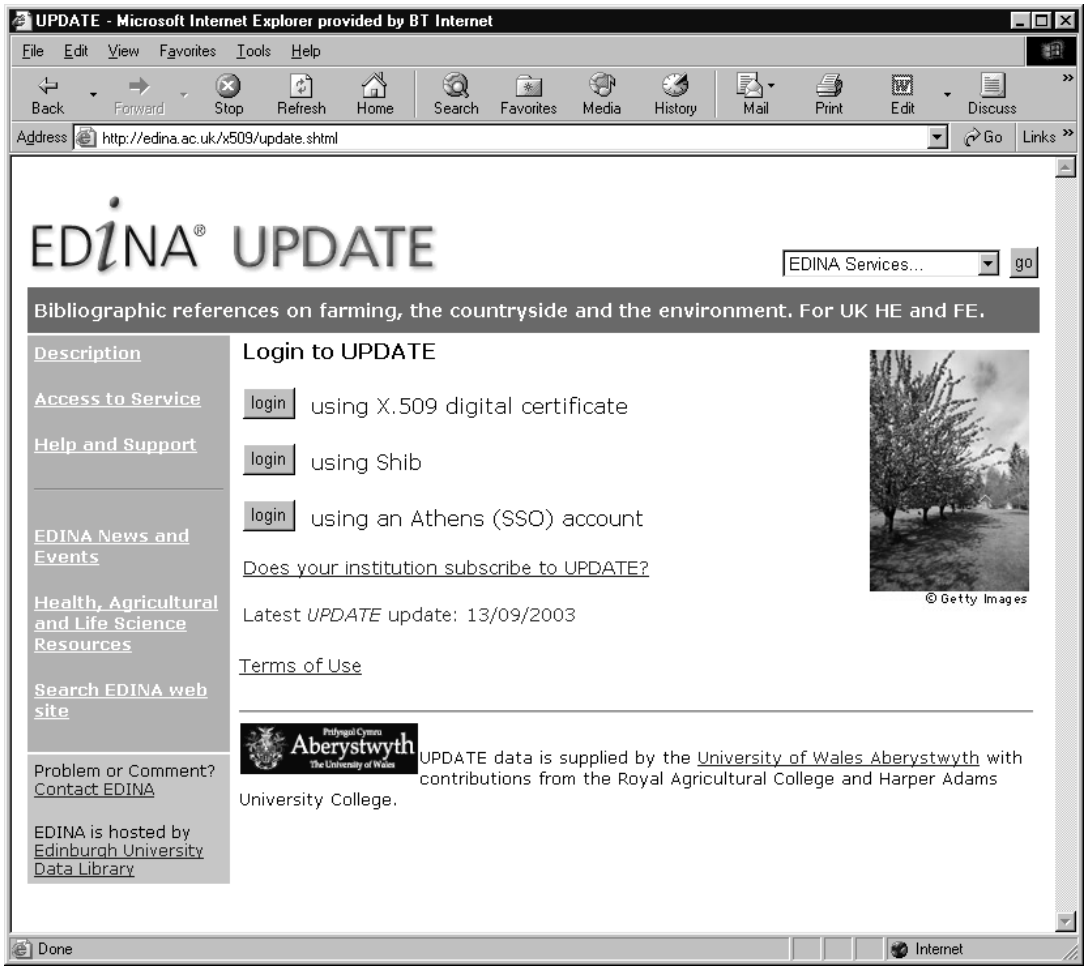
UPDATE - Microsoft Internet Explorer provided by BT Internet
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites Media History Mail Print Edit Discuss
Address http://edina.ac.uk/x509/update.shtml
Go Links
EDINA® UPDATE
EDINA Services...
go
Bibliographic references on farming, the countryside and the environment. For UK HE and FE.
Description
Access to Service
Help and Support
EDINA News and Events
Health, Agricultural and Life Science Resources
Search EDINA web site
Problem or Comment? Contact EDINA
EDINA is hosted by Edinburgh University Data Library
Login to UPDATE
login using X.509 digital certificate
login using Shib
login using an Athens (SSO) account
Does your institution subscribe to UPDATE?
Latest UPDATE update: 13/09/2003
Terms of Use
© Getty Images
Aberystwyth
UPDATE data is supplied by the University of Wales Aberystwyth with contributions from the Royal Agricultural College and Harper Adams University College.
Done
Internet

Figure A.8 — Certificate-enabled EDINA test service

A.5.3.4 Integrating SSL with session management

Given the framework described above, two approaches were considered to the problem of modifying an existing (http) service to present an SSL (https) channel to the browser for client authentication:

a) Move the existing service *en bloc* to SSL; every page gets an https address.

b) SSL-protect the login script only. Successfully authenticated users receive an HTML front page with embedded links to plain http pages as usual.

The first option is conceptually simplest, easy to implement, and might well be best for new services being set up from scratch. Its only real downside is increased load on the server: every page sent would be SSL encrypted. However, for an existing service this option would require modification of the Apache configuration file (`httpd.conf`, to create the SSL port) and possibly the web interface to the database. In order to decouple the work on certificates as much as possible from live services, this option was not pursued in our experiments.

The second option creates an https URL for the login script only. The web server behind this URL can be completely separate from the one providing the actual service (as it was in our case). The login script is a slightly modified version of the one described in A.5.3.3 above, with redundant Athens-specific functions removed. The only real drawbacks are minor. If the first page the user sees after clicking the "login with X.509 certificate" button contains embedded images from plain http addresses, as it very likely will, then old versions of Netscape (4.x) complain about the presence of "insecure" content on a secure page and replace the graphics by a "key" icon. Current versions of Netscape (7.0) and other browsers (IE, Mozilla) do not suffer from this. Additionally, most browsers pop up a message box informing users whenever they leave a secure site. Because the first page seen after clicking on the "X.509 login" button is a secure (https) page, but the pages linked from it are not, this pop-up appears when the user clicks any action button on the first screen they see after logging in, which is somewhat counter-intuitive. These issues were seen as minor niggles best addressed in the user-interface design of the web service itself (e.g., by providing a text-only login confirmation page). Option two was the one implemented in our test service, largely because of its good fit with (and decoupling from) existing software components.

A.5.3.5 Authenticating with a certificate

The "login using certificate" button added to a certificate-enabled service's front page links directly to the modified login script at an SSL-protected URL (note the https):

`https://`*`server`*`.ed.ac.uk/cgi-bin/updatelogin-x509?`

The Apache web server behind this URL is configured to require client certificates, as described in A.5.2 above. Clients without a certificate from an authority ultimately signed by the JISC national CA are rejected, and will see a standard "this page cannot be displayed" error message from their browser. Subtler error handling might be achieved by re-configuring the server to make client certificates optional and modifying the login script to reject clients without JISC-signed certificates. However, this would require a work-round for some odd behaviour in mod_ssl: the client certificate chain made available to server-side scripts via its `SSL_CLIENT_CERT_CHAIN_`*`n`* environmental variables does not include the root certificate. Without the root certificate, the script cannot determine whether or not to trust the client certificate.

NOTE: This is the second, and more significant, problem we encountered in this area of mod_ssl: the first caused all the `SSL_CLIENT_CERT_CHAIN_`*`n`* environmental variables to be empty. We traced this bug back to a literal +17 offset into a character string that should have been +18, but by the time we had done so, a fixed version was available.

A.5.3.6 Certificate login script and authorisation

```
use Edina::Autho::LoginScript;
package Edina::Autho::LoginScript;

my $use_ocsp = 1;

service("update",
        "http://server.ed.ac.uk/WebZ/Authorize?sessionid=0");

sub authorise {
  return if !$use_ocsp;
  return if ocsp_status_good();
  not_validated();
}
```

The above extract shows the gist of a login script modified for use with certificates. `Edina::Autho::LoginScript` is the same, unmodified package as used with Athens. The "`service`" call starts the named service ("`update`"), using the given URL to create a new session. Its HTML output, with embedded session identifiers, is returned as the script output.

The main point to note is that the script overrides the default "`authorise`" function provided by the `LoginScript` package. In the Athens case, the default function checks that the user has permission to access the named service. In the simplest certificate case, where OCSP revocation checking (see A.9.3 below) is not being used, the override function simply returns, indicating that the certificate is always authorised. This is valid in our test environment, and probably in some real-life situations too, because only authenticated users can get this far, and only our test JISC-signed certificates are accepted for authentication, therefore any authenticated user is automatically a member of the community authorised to use the service and no further authorisation test is required. In a more complex environment, this is the place to hang code to make explicit authorisation checks.

Examples of more sophisticated authorisation are given in A.7.2, using Athens, and A.7.3, using an external file of licensing data.

Authorisation code can be fairly simple. It need not deal directly with certificates, because mod_ssl helpfully decodes the distinguished name fields of the client certificate into environmental variables as strings, allowing code like the following, which would permit certificates issued by a particular organisation:

```
return if $ENV{SSL_CLIENT_I_DN_O} eq “University of Haddington”;
```

To simplify identifying organisations for authorisation purposes (e.g., should it be “Haddington University” above?), a TIES certification authority may include the DNS name of the issuing organisation in its certificates, encoded in Domain-name Component (DC) format:

```
SSL_CLIENT_I_DN="/CN=TIES Project Edinburgh CA/O=TIES Project at
Edinburgh/DC=ed/DC=ac/DC=uk/C=GB"
```

Bear in mind though that the DNS name itself may not be unique (e.g., ed.ac.uk and edinburgh.ac.uk are both valid) and the certificate contents are under the control of the issuer. From a data service provider’s point of view, it is probably best to keep names of authorised organisations in external tables, such as the licensing authority tables discussed in A.7.3.

Other than this authorisation function, most of the rest of the script deals with OCSP revocation checking, which is discussed in more detail in A.9.3. This code resides in the script for initial implementation convenience only. In a deployed system, it should migrate into a revised version of the `LoginScript` package.

A.5.4 DSP audit

To allow problems or misuse to be traced back to individuals, a DSP generally records the identity of each user who logs in to a site, along with a timestamp. Often the identity is just a username. When a certificate is accepted though, we want to log its serial number and the distinguished name of the subject as the user’s identity. It will also be useful to know who issued the certificate, particularly if the issuer is a regional CA or the institution itself rather than a single, national JISC CA. We therefore log the distinguished name of the issuer too. Here is an extract from the log of our test service. The issuer common name (CN) has been abbreviated (from “TIES Project Edinburgh CA”) to prevent unwanted line-breaks:

```
Tue Mar  4 11:38:28 2003: certificate #08:
   Issuer:  /CN=Edinburgh CA/O=TIES Project at Edinburgh/DC=ed/DC=ac/DC=uk/C=GB
   Subject: /CN=ext-johnm/O=TIES Project/DC=ties/DC=org/C=GB
Tue Mar  4 16:10:16 2003: certificate #04:
   Issuer:  /CN=Edinburgh CA/O=TIES Project at Edinburgh/DC=ed/DC=ac/DC=uk/C=GB
   Subject: /CN=ext-fiona/O=TIES Project/DC=ties/DC=org/C=GB
Fri Mar 14 21:17:07 2003: certificate #06:
   Issuer:  /CN=Edinburgh CA/O=TIES Project at Edinburgh/DC=ed/DC=ac/DC=uk/C=GB
   Subject: /CN=ext-mark/O=TIES Project/DC=ties/DC=org/C=GB
```

For debugging, it is also useful to record failures to log in, and the reasons for them. This is done in a separate error-log file:

```
Wed Feb 26 12:25:11 2003: certificate #03: certificate status is 'revoked'
   Issuer:  /CN=Edinburgh CA/O=TIES Project at Edinburgh/DC=ed/DC=ac/DC=uk/C=GB
   Subject: /CN=ext-fiona/O=TIES Project/DC=ties/DC=org/C=GB
```

A.6 Microsoft DSP certificate verifier

In A.4, we asked whether certificates issued by a Microsoft CA could be used to access web resources hosted on Unix by Apache servers using mod_ssl client authentication. The inverse question was also of interest: could certificates generated by an OpenSSL CA running on Unix be used to access web resources hosted on Windows by Microsoft Internet Information Server (IIS)? This question derived from the project’s requirement to develop a “Microsoft IIS component to assist institutions to deploy certificate-based security for internal services”. Since the required software is built in to IIS, this component consists solely of the documentation below, which therefore contains more details of exactly what buttons to press in which dialogue boxes than elsewhere in this report. The procedures given refer to IIS 5.0 running on Windows 2000 Server; we used Internet Explorer 6.0 and Netscape 7.0 clients under Windows 98.

A.6.1 Installing the web server certificate

Before using any of the certificate-based security features of IIS, a server certificate must be installed. This is the certificate that identifies the web server to client browsers during secure (https) sessions. The Common Name field of the certificate should match the DNS domain name of the web server machine; if it does not, the browser will warn the user. A certificate installation wizard is provided, accessed from the IIS management application (Start, Programs, Administrative Tools, Internet Services Manager). Right-clicking the default web site in the tree display then choosing *Properties* from the pop-up menu brings up the *Default Web Site Properties* dialogue box; a *Server Certificate* button on the *Directory Security* tab invokes the wizard. We had no trouble importing a certificate for this purpose created under Unix by OpenSSL.

A.6.2 Requiring client authentication

To make IIS demand certificates from its clients, press the *Edit* button in the *Secure Communications* panel of the *Directory Security* tab of the same *Default Web Site Properties* dialogue box mentioned in A.6.1 above; tick the *Require secure channel (SSL)* box and push the *Require client certificates* radio button in the *Client certificates* panel, as shown in figure A.9. After OK is pressed, IIS will reject plain http requests; it will also refuse https requests from clients without certificates.

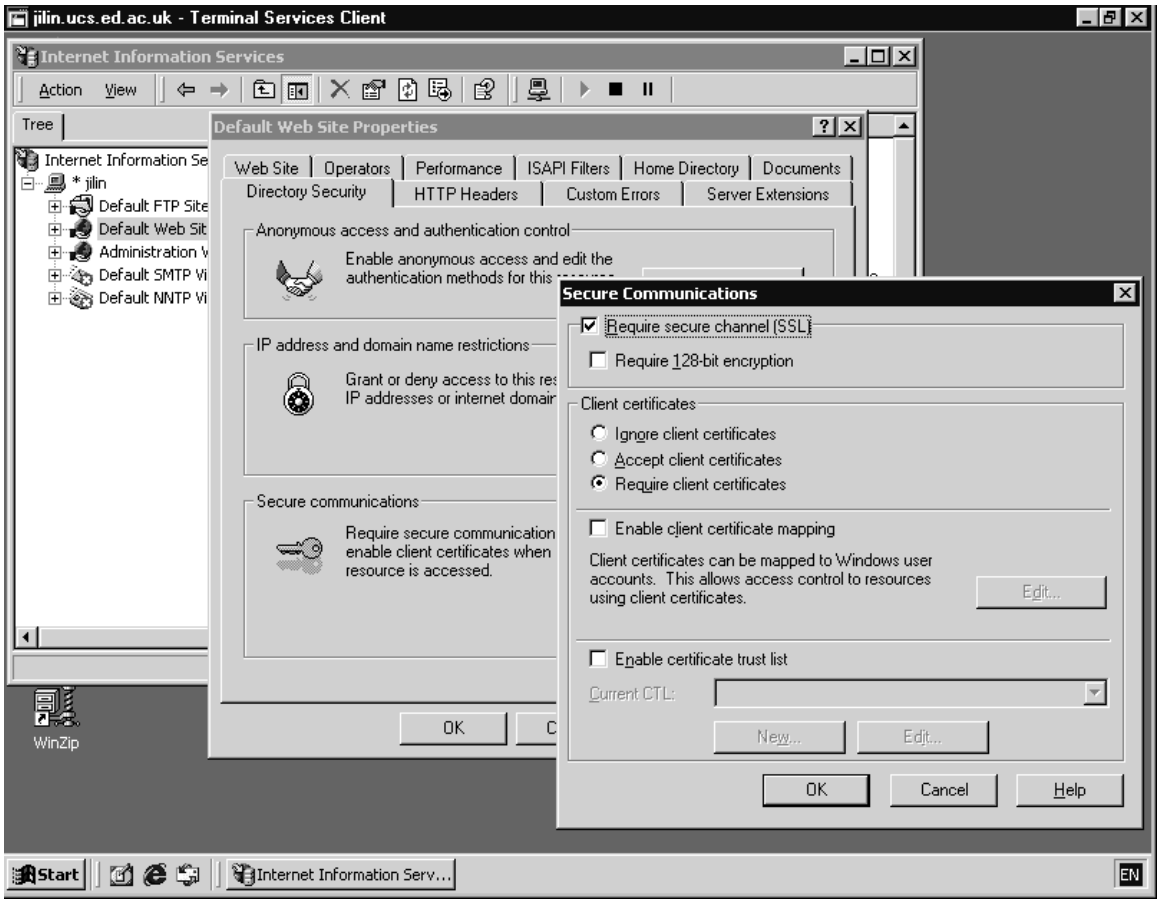

Figure A.9 — Making IIS require client certificates

A.6.3 Trusted root certification authorities

The steps followed in the previous clause configure IIS to require client certificates. However, at this point the only certificates accepted will be those ultimately signed by root CAs listed in Windows' "trusted root certification authorities" certificate store. Initially this contains only well-known commercial certification authorities and any local Certificate Services installation. Certificates signed by other authorities, in particular the JISC CA, will not be accepted. Indeed, in browsers like Netscape, certificates signed by the JISC CA will not even appear in the list of personal certificates from which the user may choose when visiting the IIS web site.

That is because, during SSL negotiation, the server tells the client browser which root CAs it is prepared to accept for client authentication; the browser presents to the user only that subset of their personal certificates.

A.6.3.1 Adding a Trusted Root CA on a Stand-Alone Machine

For a stand-alone machine, the administrator user can add the JISC CA root certificate to the trusted root certification authorities store using Internet Explorer's "Tools, Internet Options, Content, Certificates, Trusted Root Certification Authorities, Import" user interface, which invokes the certificate import wizard. After selecting the certificate file to import, choose *Place all certificates in the following store*, then *Browse.* Tick the *Show physical stores* box and navigate to *Local Computer* under *Trusted Root Certification Authorities* then click *OK.*

A.6.3.2 Adding a Trusted Root CA with Active Directory

More generally, if Active Directory is present, Microsoft treats the question of which additional CAs are to be trusted as a "group policy" issue; different policies can be applied to different sites, domains and organisational units. This is flexible but somewhat obscure. To make the JISC CA a trusted CA for the default domain, whilst logged in as the administrator user: Start, Programs, Administrative Tools, Active Directory Users and Computers. The Active Directory management console starts. Right-click the name of the site in the left-hand pane and select *Properties* from the pop-up menu. On the *Group Policy* tab, press the *Edit* button; the group policy management window appears. In the left-hand pane, navigate to: Default Domain Policy, Computer Configuration, Windows Settings, Security Settings, Public Key Policies, Trusted Root Certification Authorities. On the main *Action* menu, from *All Tasks* select *Import.* This brings up the certificate import wizard, which operates as in the stand-alone case described in A.6.3.1 above, except that the correct certificate store is known from context and offered by default this time.

A.6.4 Certificate trust lists

Once the JISC CA has been added to Windows' store of trusted root certification authorities as described above, clients with certificates issued by any institutional CA whose own certificate is signed by the JISC CA should be able to access IIS web pages. This is only half the battle though. By default, anyone with a valid certificate from any commonly recognised commercial CA will also be able to access the site.

Inappropriate access could probably be prevented in a similar way to that we used with Apache and mod_ssl (see A.5.2), by deleting all the unwanted CAs from the trusted certification authorities store. This method did not appeal, since not only is a Windows certificate store more of a system-wide entity than mod_ssl's `SSLCACertificateFile`, potentially affecting other applications too, but Windows provides a specific mechanism for restricting trust to a given list of CAs, called a Certificate Trust List (CTL).

To create a certificate trust list for an IIS web site, invoke the *Secure Communications* dialogue box described in A.6.2 and shown in figure A.9 Tick the *Enable certificate trust list* box; the greyed-out *New* button should go live. Pushing this button invokes a pretty self-explanatory wizard that allows the user to specify which root CAs are to be trusted, either by selecting certificates from the trusted root certificates store or by importing certificates from external files. The resulting certificate trust list is then given a descriptive name and saved. Multiple CTLs are permitted; the currently active one is selected from a drop-down list of the descriptive names (again, see figure A.9). We successfully set up a certificate trust list containing only the JISC CA root certificate, which filtered out client certificates from other CAs. One side-effect of the way this filtering is done is that when a client presents a certificate from an untrusted CA, the error page returned explicitly says the certificate was invalid or not trusted (compare A.5.3.5).

A.6.5 Handling of revoked certificates by IIS

A casual reader of Microsoft's documentation would be unaware even of the concept that a client certificate might be revoked. By default, IIS does not check certificates for revocation at all. Searching further, a user may discover the following document:

> `http://www.microsoft.com/windows2000/en/server/iis/default.asp?url=/windows2000/en/server/iis/htm/core/iiregglo.htm`

This reveals a global registry setting, `CheckCertRevocation`, which turns on IIS revocation checking. However:

> "[CheckCertRevocation] Specifies whether client certificates are checked for revocation by IIS. This is disabled by default because checking for common certificate issuers is typically done over the Internet and has severe performance impact. However, enabling checking may be useful if you issue your own certificates and the revocation process is local."

This is not very encouraging from a JISC information environment point of view. Further searching indicates that when checking is enabled, IIS will fetch a CRL from one of the CRL distribution point URLs embedded in a client certificate *at the time the certificate is presented* and the client tries to access a resource, i.e., on demand; see:

> http://support.microsoft.com/default.aspx?scid=kb;en-us;289749

However, a CRL fetched in this way is cached, and will not be fetched again until its expiry time. The fact that fetching the CRL takes place over the Internet when the client tries to access the certificate-protected resource is doubtless the reason for the performance warning above. In an environment with a few centralised regional or national CAs though, it is likely that in practice IIS would soon build up a complete cache of all the required CRLs, so the performance impact might not be as bad as it seems. The implications of on-demand CRL fetching in IIS for the design of a national revocation-management scheme are discussed further in A.9.4.

A.6.6 Audit

By default, IIS access logging includes both client IP address and username. The username part refers to local Windows usernames, and is shown as "-" for an "anonymous-access" web site, which is the default even when SSL secure communications with client (certificate) authentication is enabled. Logging of the client certificate to identify the end user was not investigated but would probably require writing Active Server Pages code to interrogate the client certificate and write the contents to a separate log file. The more familiar scripting provided by Apache wins out here (compare A.5.4).

Mapping client certificates to Windows usernames can be done on either a one-to-one or a many-to-one basis, by ticking the *Enable client certificate mapping* box in the *Secure communications* dialogue discussed in A.6.2 above and then filling out a series of dialogue boxes to specify pattern-matching rules for certificates. For example, one could map all certificates with an Organisation field of "University of Edinburgh" to an EDUNI user. However, this would not satisfy the requirement to backtrack to an individual end user in cases of abuse.

A.6.7 Microsoft DSP conclusion

Using the procedures described above, we were able to use certificates issued by both a Unix CA (OpenSSL) and a Windows 2000 Server CA (Microsoft Certificate Services) to access the same IIS-hosted web site. In both cases, the institutional CA certificate was signed by our (OpenSSL) JISC CA. It therefore seems as though it should be feasible for Microsoft-based portal and hybrid library systems to verify certificates, subject to the revocation issues and audit weaknesses identified in A.6.5 and A.6.6.

A.7 DSP authorisation package

Three authorisation methods were investigated:

- Shibboleth
- Athens migration
- Licensing authority tables.

A.7.1 Shibboleth

Shibboleth (http://shibboleth.internet2.edu/) is not strictly an authorization infrastructure, but rather a mechanism for allowing local authentication of users, and passing a user's authorization attributes from the home to the resource site. Whether a user is authorized or not is determined by the resource site. Shibboleth comes in two parts: the origin side that runs at an end user's institution and the target side run by the data service provider. The scope of this project covered only the target side but, inevitably, we had to implement the origin side as well, to permit testing. In fact, the origin side was implemented first, to allow us to verify our origin set-up against the

test target provided by the Shibboleth development team. Doing this first was useful, and we would recommend this approach to others deploying Shibboleth targets.

The origin is responsible for authentication, and can use whatever mechanism is appropriate to the local environment. For testing, we used Apache basic authentication, i.e., username and password from an `.htaccess` file. When a user initially browses a Shibboleth-protected site, a centrally managed Shibboleth component called the WAYF ("Where Are You From?") presents a list of institutions, from which the user selects theirs using a drop-down list box. We contacted the Shibboleth team to get our origin site included in the 0.8 development WAYF list (this process will be more automated in future versions of Shibboleth).

We installed version 0.8 of both origin and target, under Solaris 2.8, and verified that:

a) the Shibboleth team's test target correctly reported that our origin provided the (sole) attribute "`member@ed.ac.uk`";

b) our test target correctly redirected browsers to the WAYF, and when our origin site was entered there, correctly forced authentication of the user by the origin and served up a test page.

The next step was to Shibboleth-enable the same EDINA service (UPDATE) that had previously been used to test authentication directly by certificate. Another user interface button, "login using Shib", was added for this purpose (see figure A.8). This button links to a login script with similar functions to the other login scripts described in A.5.3. In this case though, the Apache web server is set to require that only clients authenticated via Shibboleth can access the script URL:

```
<Location /cgi-bin>
  AuthType shibboleth
  require valid-user
</Location>
```

As can be seen, having installed Shibboleth on the target system, protecting particular web pages (in this case, all CGI scripts) is straightforward.

The most positive aspects of Shibboleth seem to be:

a) Federated authentication (a design goal): origin sites can use different authentication schemes from the one the target site normally uses (e.g., PKI vs. Athens), provided the target site trusts the origin institution. Trust is expressed by manually granting membership of a federation and then checking that messages are optionally signed by trusted members. There can be many different federations.

b) The developers' intention to make the federation mechanism converge with the wider Liberty Alliance, http://www.projectliberty.org, which is developing federated identity mechanisms for commercial use. A Shibboleth goal is to get privacy concerns embedded into future industry standards in this area.

c) Significant momentum among US content providers. We have been following the pilot project's conference calls, and there seems to be active participation from JSTOR, OCLC, EBSCO, WebCT and Blackboard, with Elsevier not yet so involved.

d) Privacy: provision in theory for allowing end users to restrict the information their origin site may disclose about them to a target. This was another Shibboleth design goal. In practice, this provision is still rudimentary to non-existent for non-expert users, but the architecture supports it and access should continue to develop.

e) The code itself seems to be pretty robust. Most problems we have seen reported have been with requirements, building, installation and OS interaction rather than the code itself. The limited amount of direct inspection we have done confirmed our impression of good code quality.

The main drawbacks we have noticed are:

f) At the time of writing (July 2003), Shibboleth was still in a state of flux. The version we initially evaluated (0.8) was incompatible with the previous version (0.7), and, as it turned out, also with the first production release (1.0), even though 0.8 was billed as the last compatibility-breaking change. We hope this will settle down after 1.0.

g) Installation is somewhat daunting. Documentation exists and is reasonably accurate, but still resembles internals documentation more than a cookbook (this is in hand for origins). There is a lot of prerequisite

software, even for the binary installations we did (Apache, Tomcat, J2SE, mod_jk, mod_ssl, OpenSSL, exact versions of GCC shared libraries, on pain of mysterious failures), and these packages themselves are non-trivial to install. For serious production work, a source build of everything is strongly recommended, to allow timely incorporation of security patches for any of the base packages. This looked even more formidable and we did not attempt it. In practice, most Unix-based DSPs are likely to be already familiar with installation and maintenance of these packages (not all of us were), and perhaps more importantly, believe in the underlying ethos of source distribution using arcane build scripts and quaint 1970s `INSTALL` text files. The software probably does therefore meet the expectations of its target audience, but left some of us distinctly put off.

h) Shibboleth is responsible only for the transport of attributes from the user's host institution to the service target. Other substantial issues such as local authentication, attribute definition, and authorisation by attribute evaluation, are left to agents on the periphery. A simpler model could enable a service target simply to read attributes from a remote institution's LDAP directory (privacy issues should not arise concerning the public attributes of anonymous users).

i) The authorisation model is weak, and envisages only a single Source of Authority for attributes, namely the user's institution. This is the only source of attributes allowed for, in contrast to other authorisation schemes which enable authorisation decisions to be based on attributes supplied by several authorities.

j) The federated model may be well suited to US requirements but is not an obvious fit for the JISC IE where national service providers are the norm. Nor is this the model in use in the e-Science Grid. There is little evident UK requirement for a scheme that allows the establishment of multiple cross-institutional federations.

Shibboleth allows origin sites to tell targets about attributes of end users. However, this is only a mechanism for securely passing a string like "`member@ed.ac.uk`" or "`urn:mace:InCommon:entitlement:common:1`". The latter is defined by the InCommon federation to be "faculty, student, staff or library walk-in" at the origin site. Agreeing attribute semantics like this, which is important for contracts and content providers, is (necessarily) delegated to federations and is still a work in progress. Determining what a user is authorized to access, given their attributes, is left entirely up to the target site. This is where authorisation infrastructures like Akenti (http://www-itg.lbl.gov/Akenti/homepage.html) and PERMIS (http://sec.isi.salford.ac.uk/permis/) come into play. They grant or deny access according to the user's attributes and the policy of the target site.

A.7.2 Athens Migration

Currently, DSPs trust Athens to provide an authoritative source of information about which users are authorised to access which resources. It would clearly be useful to provide a means for DSPs to continue using this information during at least the early stages of a migration from Athens authentication to certificate-based authentication. Other approaches that do not depend on proprietary protocols to access authorisation information, for example the licensing authority tables discussed in A.7.3, could then be rolled out more or less at leisure.

Most JISC IE resources are licensed to institutions rather than individual users. If certificates are issued by a single central CA, the primary key for looking up authorisation status is therefore the organisation name from the subject's distinguished name. The organisation name is assumed to be taken from a controlled list maintained by the central CA. In fact, with the model of individual institutional CAs in mind, the *issuer* organisation name was used in the experiments described below but the argument is not affected.

To use Athens to determine the authorisation status of a user authenticated by a certificate, the organisation name from the certificate must be mapped to an Athens account identifier. This Athens account must be set up with rights to exactly the set of resources licensed to the institution. The EDINA certificate login script described in A.5.3.6 was modified to read this mapping from an external file, which contains lines like this:

```
University of Edinburgh|edix509,itsPassword|
```

The Athens account `edix509` must be created by the appropriate administrator using the normal Athens mechanisms. Note that every participating DSP must know the password for each account to be used in this way and that the file could in principle be maintained and distributed by the central CA. The mapping of many certificates to one Athens account via this file takes care of the mismatch between the JISC IE requirement for relatively coarse-grained authorisation and the finer-grained (per-account) control provided by Athens.

To evaluate authorisation status, the modified login script need only use standard Athens function calls. In the case of EDINA, these are conveniently encapsulated within the existing packages `Edina::Autho::LoginScript` and `Edina::Autho::AthensSSO`, so the code is quite simple. It needs to:

a) find out the IP address and DNS host name for the end user's browser (required by Athens), using existing code (`Edina::Autho::GetIPHost`);

b) hide any Athens error messages from the user by overriding the standard error handler function;

c) map the certificate's organisation name, provided by mod_ssl as a string in an environmental variable, to an Athens account and password using the external file;

d) initialise Athens with this username and password and the name of the Athens resource to be accessed, instead of the username and short-term token normally supplied (as CGI parameters) by the Athens Authentication Point web page;

e) finally, invoke the standard `Edina::Autho::LoginScript::authorised` function to evaluate authorisation status of the Athens account.

The perl code is shown below:

```
sub athens_authorised {
  my ($institution, $service) = @_;
  my ($ipaddress, $hostname) = Edina::Autho::GetIPHost::getiphost();
  $Edina::Autho::AthensSSO::athens_error_handler = \&quiet_please;
  read_licences(); # read Athens account/password file into %athensid,%athenspw
  Edina::Autho::AthensSSO::init($athensid{$institution}, # Athens username
                                $athenspw{$institution}, # password
                                $ipaddress, $hostname,
                                athens_resource_name($service));
  return 0 if $athens_err;
  $athens = 1;
  return authorised();  # standard Athens authorisation based on above user/pass
}

sub authorise {
  my $institution = $ENV{SSL_CLIENT_I_DN_O};  # issuer org. name from mod_ssl
  return if athens_authorised($institution, $service);
  bad_cert($institution." not authorised for ".uc($service));
}
```

This code was tested using standard Athens local site administration features. When the Athens resource for the test service was included in the list of authorised resources for the `edix509` account, clients holding TIES certificates could access the service. Removing access to the Athens resource, without changing either the code described above or the contents of the external file, caused the same client to be denied access.

Note that, in general, Athens offers higher-level facilities than those described above, in particular a plug-in module for Apache that encapsulates all of the authentication and authorisation work and does not require any programming. Historically, for flexibility EDINA has instead used lower-level (but published) Athens interfaces from within perl login scripts. For DSPs who use the standard plug-in, moving to certificate-based authentication would require either a new plug-in from EduServe or the adoption of a programmatic, script-based approach. The increased technical demands of the latter might not be welcomed.

As a pervasive measure, the current Athens Authentication Point web page could be updated to accept certificates and enable certificated users to access all current Athens SSO-enabled DSPs.

A.7.3 Licensing Authority Tables

Previous work [11] identified the possibility of basing authorisation decisions made by DSPs on tables maintained by a central licensing authority. These Licensing Authority Tables would list, for each resource, the acceptable authentication methods for that resource (open access, IP address checking, Athens or certificates), and, where appropriate, the set of institutions licensed to access the resource.

The test service was modified to implement a simplified version of the licensing authority tables concept. Distribution of the tables from the central authority to each individual DSP was not addressed: the table was assumed to be held within the local file system of the DSP. The format of this file was chosen to be simple for an authorisation script written in perl to parse, rather than to follow exactly the ASN.1 data model of the original paper. It is a flat text file, with each line representing one licensed institution. The organisation name string from

the issuer DN of a certificate is used to identify the institution; this name is followed by a comma-separated list of resources licensed to that institution:

```
University of Edinburgh||biosis,update
```

The empty field between the vertical bars is for the Athens migration information discussed in the previous clause; the same file is used for both. Which authorisation mechanism is used, Athens or the licensing table, is determined at run time by a switch in the code.

With the authorisation data thus separated from the login script, it was easy to verify that a certificate-authenticated user could access the test service when that service name was in the list of licensed resources but was denied access when the test service was removed from the list of licensed resources:

```
Certificate Not Accepted
University of Edinburgh not authorised for UPDATE
```

A.8 Browser support

It was anticipated at the outset of the project that since large-scale deployments of client authentication using X.509 certificates already exist in the US (University of California, University of Texas, etc.), current mainstream desktop browsers must provide adequate support. This proved to be largely true in practice, with the partial exception of the Mac platform.

Experiments were conducted on the principal desktop user platforms: Windows (98, XP), Linux 7.3, and Mac OS-X. These showed that Netscape Navigator (Mozilla) was fully functional on all platforms, with Internet Explorer also effective on Windows. The only surprise was that on the Mac neither Safari (Apple's current browser) nor IE 5.2 (also in widespread use) supported certificates. Therefore for Mac OS-X, only the Mozilla browser was functional.

One major weakness found concerns CRL processing by browsers. This is disabled by default, thus weakening the degree of confidence in the authenticity of the server the browser is accessing. The CP/CPS for the ac-PKI should mandate browser use of CRLs to ensure that some safeguard exists in the event of key compromise of trusted servers.

Platform	Browser	Result
Windows 98	IE6	OK
	Netscape 7.0	OK. Must load both institutional CA & JISC root CA certificates into browser for SSL handshake to work.
RedHat Linux 7.3	Mozilla 0.9.9	OK. Identical to Netscape 7.0 on Windows.
	Netscape 4.79	OK. As per Netscape 7.0 on Windows but refuses to show http images embedded in https pages and when loading CA certificates, user must tick at least one option (e.g., "accept for web sites") or it will silently fail. Just "OK" will do in Netscape 7.0.
	Galeon 1.2.0	FAILED. Does not support `<keygen>` tag. "Key generation in progress" pop-up appears but does not terminate.
	Konqueror 3.0.4	FAILED. Does not support `<keygen>` tag. Certificate wizard appears but gives "not implemented" error.

Table A.1 — Early cross-platform browser tests

We did not attempt to document in detail the usage procedures and peculiarities of the different browsers, since this has already been done both by their publishers and by other academic users. See, for example:

— http://www.dartmouth.edu/~pkilab/pages/More_Using_Web_Res.html

— http://www.st-andrews.ac.uk/ITS/faq/security/ca.html

We were more interested in which combinations of OS and browser version would be usable. The oldest platform we considered was Windows 98 with Internet Explorer 5 (later upgraded to IE6). This proved to be fully functional. Indeed, the certificate features of more recent Microsoft operating systems and browsers appear to have remained substantially unchanged since then from a user's point of view. The results of our initial cross-platform experiments with a test CA are shown in table A.1.

The failure of Galeon and Konqueror, both of which are based on the Mozilla code, was surprising. In both cases the cause appeared to be the `<keygen>` HTML tag, which is used to invoke client-side key generation in Netscape-derived browsers. Our initial test CA used client key generation and therefore exposed these failures (the final TIES institutional CA used central key generation).

Later in the project, after the implementation of the full institutional CA, we revisited the browser compatibility question more systematically, with a wider range of more modern platforms and browsers. These experiments demonstrated that only users holding personal certificates signed by our test institutional CA could access our test site. The results are summarised in table A.2.

Platform	Browser	Result
Windows XP	IE6	OK
	Mozilla 1.4rc2	OK
Linux	Mozilla 1.4rc2	OK
Mac OS-X	Mozilla 1.4rc2	OK
	Safari 1.0	FAILED. Does not support client certificates.
	IE5.2	FAILED. Cannot import or save new certificates.

Table A.2 — Later browser compatibility tests

The unexpected result here is the Mac, where Safari is Apple's latest default browser and IE is the traditional solution. Neither supports client certificates properly, leaving Mozilla as the functional but non-mainstream answer.

The tests undertaken were considerably more detailed than the summary shown in table A.2. In particular, for each platform and browser combination, the result of loading the browser with every possible combination of a personal certificate, our institutional CA certificate and the test JISC root-CA certificate was investigated. The full results are shown in the spreadsheet `browser-access-review.xls`. In table A.2, "OK" means that a personal certificate was required in order to gain access to the test site but some combination of the institutional and JISC CA certificates might also have to be loaded into the browser. For example, IE6 on Windows XP behaves as shown in table A.3.

Certificates installed (blank means No)			Access
Personal	Institutional CA	JISC root CA	
			Denied
		Yes	Denied
	Yes		Denied
	Yes	Yes	Denied
Yes			Denied
Yes		Yes	Denied
Yes	Yes		Granted, but server (with a certificate chain rooted in the JISC CA) shown as not verified
Yes	Yes	Yes	Granted

Table A.3 — Certificate combinations with Windows XP and IE6

This was the pattern for each of the browsers shown as "OK". In order to access the test site, at least the personal certificate and the institutional CA certificate had to be loaded. Some browsers (e.g., Netscape 4.x) also require the root CA certificate to be loaded. Usability issues were not our foremost concern, but a couple of them became apparent nonetheless:

a) When a server requests client authentication from Internet Explorer, it pops up a dialogue box asking the user to select from a list of their personal certificates. Even if a personal certificate is loaded, this list may confusingly be shown as empty if the corresponding issuer CA certificate has not also been loaded. By default, the dialogue appears even if there is only one personal certificate to choose from. However, "Tools, Internet Options, Security, Internet, Custom Level, Miscellaneous" does offer the option "don't prompt for client certificate selection when no certificates, or only one certificate exists".

b) If access is denied because no suitable client certificate is present, all the Netscape and Mozilla-derived browsers give the following error message:

   server name has received an incorrect or unexpected message. Error code –12227.

   This is almost beyond parody. Unfortunately, for the reasons discussed in A.5.3.5, it may be hard for the server to catch this error; users will therefore be left to try and decode its meaning unaided.

Finally, it may be useful to show how common certificates, like an institutional CA certificate, can be shared between multiple users of the same workstation. For Mozilla, copy the certificate database files `cert7.db` (or `cert8.db`) and `key3.db` to `C:\Program Files\Netscape\Netscape\defaults\profile\GB`. Any certificates already installed will be inherited by new user accounts created thereafter. Beware: this includes personal certificates! The system copies the centrally held profile for new users. For IE6 on Windows, CA certificates are stored in the registry. Root CA certificates are at:

```
HKEY_CURRENT_USER\Software\Microsoft\SystemCertificates\Root\Certificates\
```

Intermediate CA certificates are at:

```
HKEY_CURRENT_USER\Software\Microsoft\SystemCertificates\CA\Certificates\
```

By exporting these certificates from the `HKEY_CURRENT_USER` branch to the `HKEY_LOCAL_MACHINE` branch of the registry, they become available to all users of the machine, new and old.

A.9 Revocation

Clause 11.2 introduced the two technologies for revocation handling that we considered: CRLs and OCSP. Their implementation in Microsoft Certificate Services and IIS is described in A.4.6 and A.6.5 respectively. This clause describes how we handled revocation under Unix.

A.9.1 CRL handling at the DSP

Mod_ssl contains built-in support for CRL-based revocation checking, but it is disabled by default. Checking can be turned on using the directive `SSLCARevocationPath` in Apache's `httpd.conf` file. This simply names a directory containing PEM-encoded CRL files, which an administrator must download manually or by writing scripts. Similarly, when a CRL expires, an update must be downloaded manually or by a script. Note that the system is fail-unsafe: if no CRL is present for some CA, all its certificates will be accepted. As it stands, this mechanism is inadequate for the ac-PKI, and would require the addition of suitable automation and fail safe mechanisms before it could be considered viable. Mod_ssl accesses the CRL files via hashed names, presumably for speed. A `makefile` must therefore be run on the directory before use to create symbolic links with the necessary hashed names.

We verified that revoked certificates from an OpenSSL CA were properly rejected by mod_ssl and noted that, at least for Netscape, a clear and specific error message was presented to the end user in this case.

A.9.2 OCSP handling at the CA

The latest versions of OpenSSL contain a built-in OCSP responder that reads its certificate status information out of the flat files used by the OpenSSL built-in testbed CA implementation. Since our test CA was based on the OpenSSL testbed CA, it was straightforward to modify this OCSP responder source code to access our CA database instead of the files it expected and thus to serve up certificate status responses for our live certificate database. However, note that as of June 2003, the OCSP responder implementation in OpenSSL is strongly advised against by its publishers for production use (e.g., it is single threaded). A national deployment would need either to do work here or procure a commercial implementation.

The other step required at the CA was to modify OpenSSL's configuration file so that the certificates issued would contain the URL of our OCSP responder, allowing the DSP to find the responder without *a priori* knowledge:

```
X509v3 extensions:
    Authority Information Access:
        OCSP - URI:http://alamo.ucs.ed.ac.uk:1234/
```

A.9.3 OCSP handling at the DSP

Mod_ssl has no built-in support for OCSP. We therefore implemented OCSP revocation checking in our test service by extending its login script to invoke the OCSP client that is built into recent versions of OpenSSL, as follows:

a) Extract the client certificate from the `SSL_CLIENT_CERT` mod_ssl environmental variable and the issuing CA's certificate from `SSL_CLIENT_CERT_CHAIN_0`. Note that the latter name is hardwired and the code therefore will not work with a chain other than "client, issuer, root"; in particular a certificate directly issued by the JISC root CA ("client, root") will fail. A more flexible implementation would properly scan the whole chain.

b) Save these certificates as files named *`pid`*`.client.crt` and *`pid`*`.issuer.crt`, where *pid* is a Unix process identifier, at least theoretically allowing for concurrently executing logins, although no such load testing was performed.

c) Decode the saved client certificate as text using `openssl x509 –text` and scan it for the "`OCSP – URI`" line mentioned at the end of A.9.2 to extract the OCSP responder URL. Given the implementation in perl, a text-based approach seemed the most natural.

d) Invoke the `openssl ocsp` client, which takes the responder URL, the end user's client certificate and its issuer's CA certificate as input arguments. Fortunately, it does not require the whole of the client's certificate chain to be supplied; otherwise, the lack of a client root CA certificate from mod_ssl (see A.5.3.5) might have been an embarrassment. Presumably, the OCSP responder just looks to see if the issuer CA certificate is one of the ones it is responsible for knowing about.

e) Strangely, for a Unix program, the OCSP client does not return the certificate status indication via its return code. Instead, the script must scan the textual output for the "`Cert Status:`" line. Only if this is "`good`" will the client certificate be accepted; "`unknown`" is a possible response and will be rejected.

f) If the certificate is good, the intermediate files are deleted. If not, they are currently left behind to aid debugging efforts.

Note that with this OCSP checking code, Microsoft certificates are rejected, because they do not contain an OCSP responder URL. If required, the OCSP checks can be disabled (by setting `$use_ocsp` to 0).

A.9.4 Possible revocation schemes

OCSP and CRLs provide the technological foundations but in themselves do not fully determine the architecture of a national revocation management scheme. A number of options for such a scheme were considered:

a) *Mandatory distributed OCSP***.** If every CA were required to provide an OCSP responder and issue certificates containing the URL of that responder then a DSP could always find out whether a certificate had been revoked. This was our original concept, and is what we prototyped, but it founders if any institution deploys an out-of-the-box Microsoft CA, since these do not support OCSP. It also suffers if a small institutional CA cannot provide a high-availability OCSP responder and from the relative immaturity of OpenSSL's free responder implementation.

b) *Centralised OCSP***.** It would be possible to deploy a limited number of OCSP responders on a national or regional basis. These would regularly pull CRL updates from individual institutional CAs, which would publish the URL of the corresponding central OCSP responder in the Authority information access certificate extension. From the point of view of a DSP this scheme is identical to (a) above, except that a centrally managed OCSP responder could reasonably be expected to offer high availability. However, the consequences of a responder failing would be more widespread than in a distributed

system. The reduced number of responders would allow industrial-strength commercial implementations to be procured.

c) *On-line CRL fetch.* When presented with a certificate, the DSP there and then fetches the corresponding revocation list from a CRL distribution point URL embedded in the certificate. These CRLs may be cached by the DSP. There is clearly a major and unpredictable performance penalty if a CRL is not already in the cache, but it would be possible to prime the system beforehand. This scheme seems to be Microsoft's preferred mode of operation: support for it is built into both Certificate Services (for publishing CRLs and automatically embedding CRL distribution points in certificates) and the IIS web server (for fetching and caching CRLs, though this is turned off by default).

   However, this method does not work cleanly with Apache and mod_ssl. Mod_ssl keeps a directory of CRLs, as described in 2.2.1.1 above; matching certificates are rejected early, during client authentication. Consider a certificate from a CA for which no CRL is yet present in this directory. For static HTML pages there is no mechanism to update the directory; the certificate will simply be accepted. For dynamic content, the server-side script could conceivably download a CRL but there are open questions:

   (i) Having updated the directory, would redirecting the browser back to the same URL force re-checking for revocation or would the script have to reject the current request and let the user try again manually?

   (ii) To what extent will current or future versions of mod_ssl cache the directory in memory, thus defeating on-the-fly update attempts?

   (iii) Must Apache be re-started?

   (iv) What effect would modifying and re-hashing the directory have on concurrently executing client authentications?

d) *Distributed off-line CRL fetch.* For the reasons just outlined, off-line fetching of CRLs seems to run much more with the grain of mod_ssl than on-line fetching. Each Unix-based DSP could download nightly the CRL from every CA then re-start Apache; there would be an official list of CRL distribution points. DSPs running Microsoft IIS have no choice but to use on-line CRL fetching (option c), so all certificates would need to contain CRL distribution point information, but given this, all the CRL-based schemes should work with both Microsoft and OpenSSL CAs.

e) *Centralised off-line CRL fetch.* Similar to (d), but DSPs download CRLs from a central point rather than from individual CAs. This central service would itself need to gather CRLs from every CA, but the overall traffic seen by each CA is reduced.

f) *On-demand, distributed off-line CRL fetch.* In this variant, no central CRL service or official distribution point list is required. During the day, each DSP notes the CRL distribution point URLs from the certificates presented to it. Overnight, it downloads any new CRLs required. The drawback is that, until the next day, no revocation checking will be done on certificates from a "new" CA that has not been seen before. Pump priming could be done if required.

This is not an exhaustive list. In summary, the design of a revocation scheme will be a significant part of a national certificate rollout but some things we already know. Deployment of OCSP may be problematic if inter-operation with Microsoft CAs is required. DSPs using Microsoft IIS can either do on-line CRL fetching or no revocation checking at all, but the on-line mechanism should work with OpenSSL certificates, provided they contain CRL distribution point information. Some form of off-line fetching of CRLs looks more attractive for Unix-based DSPs. This should work with both Unix and Microsoft CAs and can coexist with IIS sites using on-line fetching.

ANNEX B –SIMPLE REGISTRATION MODEL

The simple registration model described here indicates the data exchanges within an institution, and between the institution acting as a Registration Authority (RA) and an external Certification Authority (CA). In some cases, an institution may choose to operate its own CA service rather than off-load this function to an external authority. While this may affect details of the means of communication between the RA and CA, it does not otherwise change the model presented here.

The institution, acting as RA, has responsibility for, and ownership of the reference databases of staff and students; typically, these databases are owned by Human Resources and Registry offices respectively.

This model is mainly concerned with investigation of RA functionality rather than CA operation, and is intended as a means of exploring the issues rather than as a basis for actual implementation.

B.1 Registration Authority activity

The RA is responsible for the following activities:

a) collation and management of user information obtained from the authoritative sources for staff data and student data respectively;

b) generation of additional data related to the credentials to be issued to the user;

c) conveyance of local credentials to each of the institution's local services for which the user is to be assigned an access account;

d) conveyance of external credentials to the CA responsible for issuing a digital certificate to the user;

e) conveyance of local credentials to the user by some out-of-band means, such as a sealed printed form.

The RA is also responsible for generating requests for deregistration from local services, and certificate revocation for users who cease to be recognised as members of the institution.

B.1.1 RA user records

The RA may represent information on the institution's users as follows:

```
UserRecords ::= SET OF UserRecord

UserRecord ::= SET {
    master-id           IA5String –- staff or student identifier -- ,
    personal-name       IA5String,
    local-username      LocalUsername,
    passphrase          Passphrase,
    external-username   ExternalUsername,
    validity            Validity }

LocalUsername ::=  IA5String –- may be derived from personal-name or master-id --

Passphrase ::= IA5String

ExternalUsername ::= IA5String –- will be the CN component of certificate subject name --

Validity ::= SEQUENCE {
    not-before          UTCTime,
    not-after           UTCTime }
```

In each user record, the **`master-id`** contains the master identifier assigned by the institution to the user (this may be a staff personnel number or student matriculation number or may follow some other naming convention). The **`personal-name`** contains the user's name, but is not necessarily a unique identifier (another user may have the same **`personal-name`**). The **`local-username`** together with **`passphrase`**, contains the credentials the user may use to access local institutional services. The form of the **`local-username`** will vary according to institutional practice. It may be derived from the **`personal-name`** for staff members or from the matriculation number for student members. Alternatively, it may be identical with the **`master-id`**.

For the purposes of certification, two additional items are also held. The **`external-username`** provides a pseudonymous identifier for the user which, together with the institution name, will form the name used as the

subject of the certificate. The **validity** specifies the period for which the certificate will be valid. The CA will not issue a certificate with a **validity** beyond the period it is prepared to maintain information on the certificate status.

NOTES

1. The RA obtains the authoritative list of all staff and students, extracted from the relevant databases, at regular intervals, typically daily, and identifies new users for whom credentials are to be created and users absent from the list from whom existing credentials should be withdrawn.
2. Local practice varies considerably. In some institutions, all user record information may be generated and maintained centrally; in others the responsibility for generating certain fields is devolved.
3. A user who occupies the roles of both staff member and student may be represented by distinct user records.

B.1.2 Disclosure of credentials to user

The RA conveys information to each new user specifying details of the credentials required to access any local service accounts assigned to the user.

```
LocalUserCredentials ::= SET {
    local-username      LocalUsername,
    passphrase          Passphrase }
```

The RA conveys each set of **LocalUserCredentials** to the appropriate user by means of some out-of-band process, such as a sealed printed form issued on presentation of a photo-id card.

B.1.3 Registration for local services

For each local service, the RA conveys information for each new user to whom an access account is to be assigned, and for each existing user for whom an account is to be withdrawn:

```
LocalServicesRegistration ::= SET {
    service             IA5String,
    added               SET OF LocalUserCredentials OPTIONAL,
    deleted             SET OF LocalUsername OPTIONAL }
```

The **service-name** identifies the service for which accounts are to be issued. New accounts to be created are recorded in **added**. Existing accounts to be deleted are recorded in **deleted**.

B.1.4 Registration with the Certification Authority

The RA transmits registration information to the CA at regular intervals:

```
InstitutionalRegistration ::= SET {
    institution-name    InstitutionName,
    date                UTCTime,
    register            SET OF ExternalUserRecord OPTIONAL,
    revoke              SET OF ExternalUsername OPTIONAL}

InstitutionName ::= IA5String

ExternalUserRecord ::= SET {
    external-username   ExternalUsername -- may be identical with local-username --
    passphrase          Passphrase -- identical with value in LocalUserCredentials -- ,
    validity            Validity }
```

The **institution-name** identifies the institution issuing the registration request; when combined with a user name, this forms the name used as the *subject* of the certificate. The date indicates the date and time at which the request was generated. If **register** is specified, this indicates newly registered users to whom the CA is requested to issue certificates (if requested by the user). If **revoke** is specified, this indicates a set of users whose certificates are to be revoked (if issued), or for whom subsequent certification requests are to be denied (otherwise).

For each user for whom registration is requested, an ExternalUserRecord is supplied. This contains the **external-username** that provides a pseudonymous identifier for the user, the **passphrase** the user must present with any certification request, and the **validity** period to be assigned to the certificate. The **validity** period shall not extend beyond the current academic year, plus a grace period (say, until 31 November).

Where **register** contains an ExternalUserRecord with an **external-username** that was previously registered, the new **passphrase** replaces the old. The **validity** field must be the same in both old and new records.

B.2 User/CA interaction

The user downloads his/her certificate and private key to the browser by first visiting a local institutional web Redirector where the user is instructed to install the CA's certificate on the browser. A dialog box is then presented which requests the user's **`local-username`** and **`passphrase`**. The local web Redirector looks-up the **`local-username`** in **`UserRecords`** and retrieves the corresponding **`external-username`**. The local web Redirector issues an HTTP redirection instruction to the browser containing:

— the user's **`external-username;`**

— the **`institution-name;`**

— the target URL of the CA.

The CA presents a page requesting the user's **`passphrase.`** If the **`external-username`** and **`passphrase`** pair is a match of corresponding values registered by the institution's RA, the CA issues a certificate and corresponding private-key to the user as a PKCS#12 object. The user follows the instructions on the help text to complete installation of the certificate on the browser.

NOTES

1. The local web Redirector may apply an IP address check to ensure that the local-username to external user-name mapping service is available only to users on campus.
2. The main advantage of this approach is that the user is not required to handle any additional credentials beyond those required to access local services. The same passphrase is used for both local services and certificate retrieval. To safeguard against disclosure of valid local credentials to an external agent, and to maintain user confidentiality, the local-username is never sent off-campus. However, the external-username may be used on campus for accessing local web-based services.
3. The user may download the same certificate/private key pair more than once without requiring the generation of a new certificate (and without requiring revocation of an existing certificate).

The certificate fetch request and response may be represented as follows:

```
CertificateFetchRequest ::= SET {
    institution-name    InstitutionName,
    external-username   ExternalUsername,
    passphrase          Passphrase }

CertificateFetchResponse ::= OCTET STRING -- CertificateAndKey as PKCS#12 object --
```

B.3 Recovery

Where a user loses access to his/her certificate (perhaps by forgetting the passphrase that may by used to secure it on the browser's key store), the user may repeat the certificate download request described in A.2. If the **`passphrase`** has also been lost, the user must apply for reissue of a passphrase (this should follow existing institutional practice for resetting passwords for use with local services). The institution may disclose the original passphrase or set a new passphrase, which is transmitted in a registration request to the CA.